

Modulhandbuch Smart Technology Bachelor of Science

Version B_STec25.1_W

Letzte Änderung: 2026-07-11 15:25:33

Inhaltsverzeichnis

MB001 – Analysis
MB002 – Mathematische Konzepte und Diskrete Mathematik
MB362 – Study Bootcamp Technik
MB369 – Digitaltechnik
MB327 – Digitale Schaltungen und Prozessoren
MB344 – Objektorientierte Programmstrukturen
MB358 – Study Bootcamp Informatik
MB377 – Deskriptive Statistik und Lineare Algebra
MB032 – Übertragungstechnik
MB040 – Algorithmen und Datenstrukturen
MB043 – Systemnahe Programmierung
MB048 – Elektronik
MB173 – Projekt Eingebettete Software
MB375 – Betriebswirtschaftslehre
MB057 – Fortgeschrittene Objektorientierte Programmierung
MB059 – Web-Anwendungen
MB068 – Halbleiterschaltungstechnik
MB097 – Bildbearbeitung und -analyse
MB118 – Soft Skills
MB122 – IT-Sicherheit
MB168 – Workshop Rapid Manufacturing
MB223 – Machine Learning
MB345 – Architekturen vernetzter Systeme
MB041 – Induktive Statistik
MB087 – Systemmodellierung
MB116 – Technologie der Mediengestaltung und GUI-Programmierung
MB135 – Projekt Eingebettete Systeme
MB244 – Exploratory Data Analysis
MB293 – Digital Product Management
MB356 – Projekt Connected Devices
MB366 – Datenbanken
MB373 – Robotik
MB376 – Projekt Intelligente Systeme
MB378 – Fortgeschrittene Lineare Algebra
MB379 – Konzepte der funktionalen Programmierung
MB171 – Seminar Aktuelle technologische Entwicklungen
MB172 – Laborassistenten
MB257 – Auslandssemester
MB357 – Cross Innovation Class
MB368 – Venture Lab
MB150 – Bachelor-Thesis
MB370 – Bachelor-Kolloquium
MB371 – Praktikum

Module

◆ MB001 – Analysis

Verantwortliche:	Fikret Koyuncu
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB001 – Analysis
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	90 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 120 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Fikret Koyuncu

Teilleistung:	TB002 – Übg. Analysis
Lernform:	Übung
Prüfungsform:	Teilnahme
Prüfungsdauer:	
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 15 Stunden weicht signifikant von erwarteten 2.0 ECTS ab.</i>
Lehrende:	Fikret Koyuncu

Studieninhalte:

- Bearbeitung von Übungsaufgaben aus dem Themenspektrum der zugehörigen Lehrveranstaltung
- Vorstellung und Diskussion möglicher Lösungswege

-
- Zahlentypen
 - Folgen
 - Bildungsgesetze
 - Grenzwerte
 - Funktionen, Relationen
 - Funktionstypen
 - Umkehrfunktion
 - Differentialrechnung
 - Differentiationsregeln
 - Anwendungen der Differentialrechnung (Kurvendiskussionen und Extremwerte)
 - Unendliche Reihen
 - Integralrechnung
 - Integrationsmethoden
 - Anwendungen der Integralrechnung
 - Funktionen mit zwei Variablen
 - Partielle Differentiation
 - Extremwertaufgaben ohne Nebenbedingungen

Lernergebnisse:

Die Studierenden können ...

- praktische Problemstellungen mathematisch formulieren
- beurteilen, welche analytischen Hilfsmittel zielführend sind
- neue, unklare und ungewöhnliche Aufgabenstellungen als solche erkennen und mit weiterführender Hilfestellung bearbeiten
- Lösungsansätze präsentieren und begründen

Die Studierenden ...

- kennen und verstehen die grundlegenden Begriffe, Aussagen und Methoden der Analysis,
- können mathematische Regeln korrekt anwenden,
- verstehen Beweistechniken,
- erkennen die fundamentale Bedeutung des Grenzwertbegriffes für die Analysis,
- beherrschen die Methoden des Differenzierens und Integrierens,
- können die eindimensionale Differentialrechnung bei praxisorientierten Fragestellungen flexibel in unterschiedlichen Fachgebieten einsetzen und dabei beurteilen, welche analytischen Hilfsmittel für welche Problemstellungen zielführend sind
- erkennen die Anwendbarkeit und den Nutzen der Analysis für unterschiedliche Fachgebiete und deren spezifischen Problemstellungen,
- können praxisorientierte Problemstellungen in mathematische Beziehungen bzw. Modelle umzusetzen und anhand analytischer Modelle weiter bearbeiten
- können neue, unklare und ungewöhnliche Aufgabenstellungen als solche erkennen und zur Bearbeitung weiterführende Hilfestellung in Anspruch nehmen,
- verfügen über gesteigerte Kompetenzen sich Fähigkeit durch Selbststudium anzueignen und sich in neue formale Systeme einzuarbeiten

Voraussetzungen und Empfehlungen:

- Schulbildung in mathematischen Grundlagen
- Empfehlung: Brückenkurs Mathematik

Literatur:

- BÖHME, Gert:
Analysis 1.
6. Aufl. Berlin: Springer-Verlag, 1990
- FETZER, Albert; FRÄNKEL, Heiner:
Mathematik 1.
10. bearbeitete Aufl. Berlin: Springer-Verlag, 2008
- FETZER, Albert; FRÄNKEL, Heiner:
Mathematik 2.
6. korrigierte Aufl.. Berlin: Springer-Verlag, 2009
- HENZE, Norbert; Last, Günter:
Mathematik für Wirtschaftsingenieure 1.
2. Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2005
- KUSCH, Lothar:
Mathematik. Aufgabensammlung mit Lösungen. Bd. 3
9. Aufl. Berlin: Cornelsen Verlag, 1995
- OHSE, Dietrich: Mathematik für Wirtschaftswissenschaftler 1. Analysis.
6. Aufl. München: Verlag Vahlen, 2004
- PAPULA, Lothar:
Mathematik für Ingenieure und Naturwissenschaftler 1: Ein Lehr- und Arbeitsbuch für das

Grundstudium.

12. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2009

- PREUSS, Wolfgang; WENISCH, Günter:
Lehr- und Übungsbuch Mathematik 1: Grundlagen - Funktionen - Trigonometrie.
2. neu bearbeitete Aufl. München: Carl Hanser Verlag, 2003
- PREUSS, Wolfgang; WENISCH, Günter:
Lehr- und Übungsbuch Mathematik 2: Analysis.
3. Aufl. München: Carl Hanser Verlag, 2003
- PAPULA, Lothar: Mathematik für Ingenieure und Naturwissenschaftler: Klausur- und Übungsaufgaben 4. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2010

Studiengänge:

- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB002 – Mathematische Konzepte und Diskrete Mathematik

Verantwortliche:	Sebastian Iwanowski
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB003 – Diskrete Mathematik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	5.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 120 Stunden weicht signifikant von erwarteten 5.0 ECTS ab.</i>
Lehrende:	Sebastian Iwanowski

Studieninhalte:

- Logik
 - Einführung
 - Aussagenlogik
 - Prädikatenlogik
- Mengenlehre
 - Grundlegende Begriffe und Konzepte
 - Relationen
 - Funktionen
 - Boolesche Algebren
- Beweisführung
 - Strukturen der mathematischen Beweisführung
 - Vollständige Induktion
 - Beweisstrategien
- Zahlentheorie
 - Teilbarkeit
 - Teilen mit Rest
 - Primzahlen
 - Modulare Arithmetik
- Algebraische Strukturen
 - Gruppen
 - Körper
- Kombinatorik
 - Zählformeln für Mengen
 - Permutationen
- Graphentheorie
 - Terminologie und Repräsentation
 - Wege in Graphen
 - Bäume
 - Planare Graphen
 - Färbungen

Lernergebnisse:

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Beherrschen der grundlegenden mathematischen Begriffe und Konzepte (Definition, Satz, Beweis) und Fähigkeit zur Unterscheidung derselben.
- Beherrschen der Grundlagen und der Formalisierung logischen Denkens.
- Verständnis elementarer Logik und Mengenlehre und des inneren Zusammenhangs dieser Gebiete.
- Darauf aufbauendes Verständnis von Relationen und Funktionen.
- Fähigkeit, elementare Beweisprinzipien wie vollständige Induktion in verschiedenen Kontexten anzuwenden.
- Beherrschen der grundlegenden Sätze der elementaren Zahlentheorie, Gruppen- und Körpertheorie, Kombinatorik und Graphentheorie und selbständige Anwendung an Beispielen.

Voraussetzungen und Empfehlungen:

Mathematik Gymnasium 9. Klasse

Literatur:

- Sebastian Iwanowski / Rainer Lang:
Diskrete Mathematik mit Grundlagen, Springer 2014, ISBN 978-3-658-07130-1 (Print), 978-3-658-07131-8 (Online)
- Albrecht Beutelspacher / Marc-Alexander Zschiegner:
Diskrete Mathematik für Einsteiger.
Vieweg 2004 (2. Auflage), ISBN 3-528-16989-3
- Norman L. Biggs:
Discrete Mathematics.
Oxford University Press 2002, ISBN 0-19-850717-8
- Neville Dean: Diskrete Mathematik.
Pearson Studium, Reihe "im Klartext" 2003, ISBN 3-8273-7069-8
- Christoph Meinel / Martin Mundhenk:
Mathematische Grundlagen der Informatik.
Teubner 2002 (2. Auflage), ISBN 3-519-12949-3

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB362 – Study Bootcamp Technik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	2 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB356 – Study Bootcamp Technik - Projekt
Lernform:	Projektarbeit
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	75 Min.
ECTS:	8.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	34 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	148 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 268 Stunden weicht signifikant von erwarteten 8.0 ECTS ab.</i>
Lehrende:	Jürgen Günther Ulrich Hoffmann

Teilleistung:	TB393 – Study Bootcamp Technik - Ingenieurskompetenz
Lernform:	Bootcamp
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	7.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	18 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	78 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	125 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	213 Stunden
Lehrende:	Dominik Miller Carsten Burmeister Andreas Haase Jürgen Günther

Studieninhalte:

Kreativitätstechniken und Brainstorming (Crazy 8s, Mind-Mapping); Grundlagen des Projektmanagements (Kanban, Scrum); Grundlagen der Programmierung für Mikrocontroller (MicroPython auf ESP32); Digitale Konstruktion (CAD) für Rapid Prototyping (FDM-3D-Druck, Laser-Cutter); Technisches Zeichnen; Grundlagen der Elektrotechnik; Grundlagen der Mechanik; Physikalisches Praktikum. Das Modul wird durch ein 6-wöchiges interdisziplinäres Teamprojekt (4 Personen) strukturiert, in dem ein smartes physisches Gerät konzipiert, entworfen und gefertigt wird. Wöchentliche Plenarversammlungen dienen der Fortschrittspräsentation; das Semesterergebnis wird abschließend öffentlich präsentiert.

Lernergebnisse:

Im Feld „Projektentwicklung“ können die Studierenden ...

- Ideen für ein stoffliches Produkt entwickeln und bewerten.
- eine Produktidee strukturiert konkretisieren und in eine umsetzbare Aufgabenstellung überführen.
- einen Prototypen durch technische Zeichnungen planen und praktisch fertigen.
- Projektarbeit koordinieren, Projektfortschritt dokumentieren und zielgruppengerecht präsentieren.

Im Feld „Technisches Zeichnen“ können die Studierenden ...

- technische Zeichnungen lesen und verstehen.
- einfache normgerechte Zeichnungen selbst erstellen.
- Toleranzen, Passungen und Oberflächengüten interpretieren.

- abhängig von der Zeichnung geeignete Bauteile auswählen.

Im Feld „CAD“ können die Studierenden ...

- grundlegende CAD-Funktionen sicher anwenden.
- sich selbständig in weitergehende CAD-Funktionen einarbeiten.
- Dateien für den 3D-Drucker und Laser-Cutter vorbereiten und exportieren.
- normgerechte CAD-Zeichnungen erstellen.

Im Feld „Programmierung“ können die Studierenden ...

- einfache Programme in MicroPython strukturieren und auf einem ESP32 ausführen.
- digitale und analoge Ein- und Ausgaben eines Mikrocontrollers ansteuern.
- Sensoren und Aktoren programmatisch einbinden und im Projektkontext nutzen.
- Fehler im Code systematisch identifizieren und beheben.

Im Feld „Elektrotechnik“ können die Studierenden ...

- grundlegende elektrische Größen (Spannung, Strom, Widerstand, Leistung) benennen und deren Zusammenhänge erläutern.
- einfache Gleichstromschaltkreise analysieren und dimensionieren.
- Schaltpläne lesen und einfache Schaltungen auf einem Steckbrett aufbauen und testen.
- typische elektronische Bauelemente (Widerstände, LEDs, Transistoren, Sensoren) identifizieren und fachgerecht einsetzen.

Im Feld „Mechanik“ können die Studierenden ...

- grundlegende mechanische Größen (Kraft, Drehmoment, Druck, Reibung) benennen und in einfachen Systemen anwenden.
- statische und kinematische Zusammenhänge an technischen Bauteilen und Konstruktionen qualitativ beurteilen.
- geeignete Materialien und Fertigungsverfahren für einfache mechanische Komponenten auswählen und begründen.
- die mechanische Machbarkeit eines Entwurfs anhand grundlegender Überlegungen einschätzen.

Im Feld „Physikalisches Praktikum“ können die Studierenden ...

- physikalische Messungen systematisch planen, durchführen und protokollieren.
- Messergebnisse auswerten, grafisch darstellen und interpretieren.
- Messunsicherheiten abschätzen und im Ergebnis berücksichtigen.
- den Bezug zwischen physikalischen Grundgesetzen und ihrer praktischen Relevanz im ingenieurtechnischen Kontext herstellen.

Voraussetzungen und Empfehlungen:

Keine formalen Voraussetzungen; das Modul ist explizit als Einstieg in das Studium konzipiert. Neugierde, Teamfähigkeit und die Bereitschaft zum praktischen Ausprobieren und zur öffentlichen Präsentation von Ergebnissen werden vorausgesetzt.

Literatur:

- Monk, S.: *Programming ESP32: Learn MicroPython Coding and Electronics*. MonkMakes, 2nd ed., 2024. ISBN 978-1-73948-745-4.
- Giesecke, F. E. et al.: *Technical Drawing with Engineering Graphics*. 15th ed. Pearson, Hoboken, 2016. ISBN 978-0-13-401152-5.

- Tipler, P. A., Llewellyn, R. A.: *Physik für Wissenschaftler und Ingenieure*. 7. Aufl. Springer Spektrum, Berlin, 2015. ISBN 978-3-642-54165-0.

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB369 – Digitaltechnik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB069 – Prakt. Digitaltechnik
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	30 Min.
ECTS:	2,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	4 Termine
Zeit in Veranstaltungen:	10 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	4 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	46 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Thomas Starke

Teilleistung:	TB364 – Digitaltechnik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 75 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Einleitung: Digitale Systeme
- Mathematische Grundlagen
 - Entstehungsgeschichte
 - Aussagenlogik und Boolesche Algebra
 - Schaltalgebra, Schaltfunktionen und Schaltfunktionssysteme
 - Operatorensysteme
 - Normalformen und Dualitätsprinzip
- Schaltnetze
 - Darstellung
 - Vereinfachung (KV-Diagramme, QMCV, BDDs)
 - Analyse (Funktion, Komplexität, Zeitverhalten)
 - Synthese und Realisierung
 - Beispiele
- Speicherelemente
- Schaltnetzentwurf: Schaltnetz wird aus einer gegebenen Spezifikation formal entworfen. Der Entwurf wird auf einem IC-Trainer realisiert. Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. Die Ergebnisse werden dokumentiert.
- Schaltwerkentwurf: Schaltwerk (z. B. ein Zähler) wird aus einer gegebenen Spezifikation formal entworfen. Der Entwurf wird auf einem IC-Trainer realisiert. Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. Die Ergebnisse werden dokumentiert.

Lernergebnisse:

Die Studierenden ...

- kennen die mathematischen Grundlagen des Schaltnetz-Entwurfs
- können darauf aufbauend einfache Schaltnetze entwerfen und optimieren
- erkennen Schaltnetze als technische Umsetzung von Schaltfunktionen
- kennen grundlegende Bausteine digitaler Systeme (Logikgatter, Multiplexer, Demultiplexer, arithmetische Schaltungen)
- können die Funktionsweise von einfachen Speicherelementen erläutern
- sind fähig, einfache digitale Systeme zu begreifen, zu spezifizieren, zu entwerfen und zu optimieren

Voraussetzungen und Empfehlungen:

Grundlegende Vorstellung von digitalen Rechnern

Literatur:

- Hoffmann, Dirk: Grundlagen der technischen Informatik, 5. Auflage, Carl Hanser Verlag 2016
- Schiffmann, Wolfram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Beuth, Klaus: Elektronik 4. Digitaltechnik, 13. Auflage Vogel Verlag und Druck 2003

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB327 – Digitale Schaltungen und Prozessoren

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB319 – Digitale Schaltungen, Prozessoren
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	150 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	40 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 105 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki Dennis Säring

Studieninhalte:

- Rechnerarchitekturen (Entwicklung, Architekturkonzepte, parallele und nicht sequentielle Architekturen)
- Mikroprogrammierung
- Architekturkonzepte
- Mikroprogrammierung
- Mehrprozessorsysteme
- Aktuelle und zukünftige Entwicklungen
- Schaltwerke
 - Einleitung und Grundbegriffe, Definitionen
 - Speicherelemente
 - Analyse
 - Synthese
 - Zusammenschaltung
 - Transformationen
 - Zustandskodierung
 - Zustandsminimierung
 - Realisierung, Beispiele
- Zeitverhalten
 - Zeitverhalten von Schaltnetzen
 - Modellierung der Gatter- und Leitungsverzögerungen
 - Statische Timing-Analyse (STA)
 - Zeitverhalten von Schaltwerken
 - Metastabilität

Lernergebnisse:

Die Studierenden ...

- kennen grundlegende Rechnerarchitekturkonzepte, die beschreiben, wie verschiedene Baugruppen von Rechnern zusammenarbeiten und wie sich unterschiedliche Rechnersysteme voneinander unterscheiden
- kennen die Funktionselemente von Rechnern mit ihren typischen Systemeigenschaften und deren Abbildung auf ein bestimmtes Architekturmodell

- können das Zusammenwirken der beteiligten Hardware- und Softwarekonzepte im Rahmen einer Aufgabe zur Informationsverarbeitung einschätzen
- besitzen ein Verständnis für Ansätze zur Steigerung der Systemleistung insbesondere unter Berücksichtigung der Aspekte von Parallelität
- kennen Aufbau und Funktionsweise von Speicherelementen und Schaltwerken
- beherrschen die Methoden zur Analyse, Darstellung und Vereinfachung von endlichen Zustandsautomaten
- erkennen ein Schaltwerk als technische Umsetzung eines endlichen Zustandsautomaten
- beherrschen die Methoden der Zeitverhaltensanalyse und Zeitverhaltensoptimierung von digitalen Systemen, sowie können das Zeitverhalten und die Zeitvorgaben beim Entwurf digitaler Systemen berücksichtigen
- können digitale Systeme mittlerer Komplexität begreifen, spezifizieren, entwerfen und optimieren

Voraussetzungen und Empfehlungen:

Kenntnisse der wichtigsten Komponenten eines Rechnersystems.

Literatur:

- Martin: Einführung in die Rechnerarchitektur, Fachbuchverlag Leipzig, 2003
- Oberschelp, Gossen: Rechneraufbau und Rechnerstrukturen, Verlag Oldenbourg 1998
- van de Goor: Computer Architecture and Design, Verlag Addison Wesley, 1989
- Müller-Schloer, Schmitter: RISC-Workstation Architekturen, Verlag Springer 1991
- Ungerer: Datenfluß-Rechner, Verlag Teubner, 1993
- Hoffmann, Dirk: Grundlagen der technischen Informatik, Carl Hanser Verlag 2007
- Schiffmann, Wolram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Rabaey, Jan; Chandrakasan, Anantha; Nikolic, Borivoje: Digital Integrated Circuits, A Design Perspective, 2nd edition, Prentice Hall 2003
- Beuth, Klaus: Elektronik 4. Digitaltechnik, 13. Auflage, Vogel Verlag und Druck 2003

Studiengänge:

- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB344 – Objektorientierte Programmstrukturen

Verantwortliche:	Dennis Proppe
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB338 – Objektorientierte Programmstrukturen
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	120 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	15 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Dennis Proppe

Teilleistung:	TB383 – Übg. Objektorientierte Programmstrukturen
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	3.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	22 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	50 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	82 Stunden
Lehrende:	Gerit Kaleck

Studieninhalte:

Es wird in die Programmierung mit Java und die Entwicklungsumgebung IntelliJ eingeführt. In der Übung werden die in der Vorlesung vorgestellten Grundkonzepte der objektorientierten Programmierung durch das Lösen verbal formulierter Aufgabenstellungen in kleinen Teams angewendet. Das Testen und Präsentieren sauber strukturierter Lösungen wird geübt.

Behandelte Grundkonzepte sind:

- Grundkonzept der Programmiersprache Java
 - Grundlegende Eigenschaften der Sprache
 - Grundlegender Aufbau von Java-Programmen
 - Ausführung von Java-Programmen
- Grundlegende Programmelemente
 - Primitive Datentypen in Java
 - Variablen, Zuweisung, Gültigkeitsbereiche
 - Operatoren und Ausdrücke
 - Anweisungen
- Referenzdatentypen
 - Arrays
 - Klassen
- Statische Methoden
- Grundlegende Klassen
 - String
 - StringBuilder
 - Wrapper-Klassen für primitive Datentypen
 - Enum
- Grundkonzepte der Objektorientierung
 - Klassen und Instanzen mit Attributen und Methoden
 - Sichtbarkeit, Packages
 - Konstruktoren
 - Vererbung und Überschreiben
 - Dynamisches Binden, Polymorphie
 - Objektorientierte Realisierung rekursiver dynamischer Datenstrukturen (Listen)
 - Generische Typen
 - Abstrakte Klassen und Interfaces - Deklaration und Nutzung
 - Realisierung grafischer Benutzungsoberflächen

- Behandlung von Laufzeitfehlern
- Klassen zur Realisierung von Dateioperationen

Lernergebnisse:

Die Studierenden ...

- identifizieren die Basiskonzepte der Objektorientierten Programmierung und stellen diese den Konzepten der prozeduralen Programmierung gegenüber.
- entwickeln Software auf der Grundlage der Kernkonzepte der Objektorientierten Programmierung.
- stellen die grundlegenden Sprachelemente (Datentypen, Anweisungen, Realisierung von objektorientierten Konzepten) von Java zusammen und wählen daraus aus, um Java-Programme mittlerer Komplexität zu entwickeln.
- vergleichen die Programmiersprachen Pascal und Java und stellen ihre Gemeinsamkeiten und Unterschiede heraus.
- setzen eine moderne Entwicklungsumgebung zur Unterstützung der Softwareentwicklung ein und stellen die damit verbundenen Funktionalitäten und Vorgehensweisen dar.
- entwerfen einfache dynamische Datenstrukturen im Kontext einer objektorientierten Programmiersprache.
- erläutern grundlegende Algorithmen, die auf den vermittelten Datenstrukturen arbeiten.
- entwerfen für Programme mittlerer Komplexität durch Einsatz geeigneter Elemente der Programmiersprache Java eine angemessene Modularisierung und legen entsprechende Schnittstellen zwischen den Modulen fest.
- benennen die Grundregeln der benutzungsgerechten Gestaltung von Programmen und nutzen diese, um Benutzungsoberflächen von Programmen begrenzter Funktionalität sowohl strukturell als auch funktional angemessen zu gestalten.
- kennen die grundlegenden Klassen und ihre Operationen, mit denen dateibezogene Operationen implementiert werden können.

Die Studierenden ...

- kennen die Basiskonzepte objektorientierter Programmiersprachen und können sie in Java umsetzen.
- können einfache dynamische Datenstrukturen im Kontext einer objektorientierten Programmiersprache umsetzen und grundlegende Algorithmen auf diesen Datenstrukturen anwenden.
- sind firm in Nutzung einer aktuellen Version einer verbreiteten Entwicklungsumgebung (IntelliJ).
- können ein vollständiges Software-System kleineren Umfangs ausgehend von einer verbalen Aufgabenstellung realisieren.
- entwickeln Software erfolgreich im kleinen Team.
- ermitteln geeignete Testfälle zur Qualitätssicherung.
- kennen die Grundregeln zur Gestaltung benutzungsgerechter Oberflächen und bedienfreundlicher Software.

Voraussetzungen und Empfehlungen:

Die im intoCoding-Bootcamp vermittelten Konzepte sollten verstanden sein und flüssig umgesetzt werden können. Die Installation der Entwicklungsumgebung IntelliJ auf dem eigenen Rechner ist empfehlenswert.

Literatur:

- Christian Ullenboom: Java ist auch eine Insel. 17. Auflage, Rheinwerk Verlag, 2023
- Hans-Peter Habelitz: Programmieren lernen mit Java. 7. Auflage, Rheinwerk Computing, 2022

- Michael Bonacina: Java Programmieren für Einsteiger: Der leichte Weg zum Java-Experten! 2. Auflage, BMU Verlag, 2018
- Markus Neumann: Java Kompendium: Professionell Java programmieren lernen. BMU Verlag, 2019
- Dietmar Ratz et al.: Grundkurs Programmieren in Java. 8. Auflage, Carl Hanser Verlag, 2018
- Michael Inden: Einfach Java: Gleich richtig programmieren lernen. dpunkt.verlag, 2021
- David Kopec: Algorithmen in Java, 32 Klassiker vom Rucksackproblem bis zu neuronalen Netzen, 1. Aufl. Rheinwerk Computing, 2021
- Kathy Sierra et al.: Java von Kopf bis Fuß: Eine abwechslungsreiche Entdeckungsreise durch die objektorientierte Programmierung. O'Reilly, 2023
- Ralph Steyer: Einführung in JavaFX/OpenJFX: Moderne GUIs für RIAs und Java-Applikationen. 2. Aufl., Springer Vieweg, 2022
- Anton Epple: JavaFX 8: Grundlagen und fortgeschrittene Techniken. dpunkt.verlag, 2015
- Sergey Grinev: Mastering JavaFX 10: Build advanced and visually stunning Java applications. Packt Publishing, 2018
- Herbert Schildt: Introducing JavaFX 8 Programming (Oracle Press). Mcgraw-Hill Education, 2015

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB358 – Study Bootcamp Informatik

Verantwortliche:	Dennis Proppe
Moduldauer:	2 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB352 – Study Bootcamp Informatik
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	60 Min.
ECTS:	15,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	32 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	140 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	90 Stunden
Aufwand während Semesterferien:	40 Stunden
Flexibel einteilbarer Aufwand:	200 Stunden
Gesamtaufwand:	470 Stunden
Lehrende:	Dennis Säring Lars Neumann Malte Heins Dennis Proppe

Studieninhalte:

- Imperative Programmierung: Dies ist der Kern des Moduls. Systematische Einführung in das algorithmische Denken und die strukturierte Problemlösung. Vermittlung der fundamentalen Konzepte der Programmierung anhand einer modernen Programmiersprache, inklusive EVA-Prinzip, Variablen, elementare und komplexe Datentypen, Kontrollstrukturen (Sequenz, Selektion, Iteration), Funktionen und Modularisierung. Entwicklung von Algorithmen und einfachen Datenstrukturen und deren Darstellung z.B. mittels Struktogrammen.
- Grundlagen der Systembedienung und Shell-Programmierung (Unix): Einführung in die Arbeit auf der Kommandozeile. Vermittlung der Navigation im Dateisystem, der Verwaltung von Dateien und Prozessen sowie der Benutzerrechte. Grundlagen der Shell-Programmierung zur Automatisierung von Aufgaben mittels Skripten, Pipes und Umleitungen.
- Grundlagen der Informationstechnik und der Informationstheorie: Vermittlung eines fundamentalen Verständnisses für den Aufbau und die Funktionsweise digitaler Systeme. Dies umfasst die Geschichte der Informationstechnik, die Von-Neumann-Architektur, die Kernkomponenten eines Rechners (CPU, Speicher, Ein-/Ausgabe), Grundlagen von Schaltnetzen sowie die theoretischen Grundlagen der Datenrepräsentation (Binär- und Hexadezimalsystem, Zweierkomplement).
- Praktische Programmierprojekte: Kontinuierliche Anwendung und Vertiefung der theoretischen Inhalte durch zahlreiche praktische Übungen, Programmieraufgaben, begleitende Programmiertests

Lernergebnisse:

Nach erfolgreichem Abschluss des Moduls "Study Bootcamp Programmieren" besitzen die Studierenden folgende Kompetenzen:

Fachkompetenzen:

- Systembedienung: Einen Computer sicher und effizient über die Kommandozeile zu bedienen und einfache Abläufe durch Shell-Skripte zu automatisieren.
- Algorithmisches Denken: Problemstellungen zu analysieren, in logische Teilschritte zu zerlegen und dafür systematisch Algorithmen zu entwickeln.
- Programmierung: Die grundlegenden Konzepte der imperativen Programmierung zu verstehen und diese praktisch in einer modernen Programmiersprache umzusetzen, um lauffähige

Programme zu erstellen.

- Code-Qualität: Strukturierten, lesbaren und kommentierten Code zu verfassen und grundlegende Debugging-Strategien anzuwenden.
- IT-Architektur: Den prinzipiellen Aufbau eines Computers und die grundlegende Funktionsweise von Rechnernetzen zu erläutern.
- Zahlensysteme: Sicher zwischen verschiedenen Zahlensystemen (Dezimal, Binär, Hexadezimal) umzurechnen und die binäre Repräsentation von Daten zu verstehen.

Methodenkompetenzen:

- Problemlösung: Komplexe Aufgabenstellungen systematisch zu erfassen und in eine programmiertechnische Lösung zu überführen.
- Abstraktionsvermögen: Konkrete Anforderungen in abstrakte Datenstrukturen und logische Abläufe zu übersetzen.
- Selbstgesteuertes Lernen: Eigenständig Problemlösungen zu recherchieren und sich mithilfe von Dokumentationen und anderen Ressourcen in neue technische Sachverhalte einzuarbeiten.
- Fehleranalyse: Systematisch Fehler im eigenen Code zu identifizieren, zu analysieren und zu beheben (Debugging).

Sozial- und Personale Kompetenzen:

- Teamarbeit: In kleinen Teams (z.B. via Pair Programming) effektiv zusammenzuarbeiten, um gemeinsam Programmieraufgaben zu lösen.
- Kommunikation: Eigene Lösungsansätze und Code-Strukturen verständlich zu erklären und technische Sachverhalte präzise zu diskutieren.
- Selbstorganisation & Zeitmanagement: Die Bearbeitung von Programmieraufgaben und Projekten eigenverantwortlich zu planen und Meilensteine termingerecht zu erreichen.
- Frustrationstoleranz: Resilienz und Ausdauer bei der systematischen und oft langwierigen Suche nach Fehlern in Software zu zeigen.

Voraussetzungen und Empfehlungen:

Interesse für Informatik, Computer und Softwareentwicklung.

Literatur:

Für Informationstechnik:

Gumm, Hans-Peter; Sommer, Manfred: Einführung in die Informatik, Oldenbourg, 8. Auflage 2009.

Müller, Käser, et., al.: Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003

Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag 1998

Martin: Einführung in die Rechnerarchitektur, Fachbuchverlag Leipzig, 2003

Für Go:

Bodner, J. (2021). Learning Go: An idiomatic approach to real-world Go programming. O'Reilly Media.

McGavren, J. (2019). Head First Go: A brain-friendly guide (1st ed.). O'Reilly Media.

Harsanyi, T. (2022). 100 Go mistakes and how to avoid them. Manning Publications Co.

Für Unix:

Kofler, Michael: Linux: Das umfassende Handbuch, Rheinwerk Computing, 2023, ISBN: 978-3836284424

Dietze, Martin: Praxiskurs Unix-Shell, O'Reilly Verlag GmbH & Co. KG; 2011, ISBN: 978-3897215658

Robbins, Arnold; Beebe, Nelson H.F.: Klassische Shell-Programmierung, O'Reilly Verlag GmbH & Co. KG; 2006, ISBN: 978-3897214415

Kernighan, Brian W.; Pike, Rob: UNIX-Werkzeugkasten: Programmieren mit UNIX, Hanser Fachbuch, 1986, ISBN: 978-3446142732

Friedl, Jeffrey E. F.: Reguläre Ausdrücke, O'Reilly Verlag GmbH & Co. KG, 2007, ISBN: 978-3897217201

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB377 – Deskriptive Statistik und Lineare Algebra

Verantwortliche:	Franziska Bönte Andreas Haase
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB373 – Deskriptive Statistik, Lineare Algebra
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	5.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 135 Stunden weicht signifikant von erwarteten 5.0 ECTS ab.</i>
Lehrende:	Franziska Bönte Andreas Haase

Studieninhalte:

Im Rahmen der linearen Algebra werden folgende Themen behandelt:

- Vektorrechnung
 - Geometrische und arithmetische Vektoren
 - Rechenregeln
 - Lineare (Un-)Abhängigkeit
- Lineare algebraische Gleichungssysteme
 - Gauß-Algorithmus
 - Systematisierung des Lösungsverhaltens
 - Unterbestimmte Systeme
- Matrixrechnung
 - Matrixalgebra
 - Inverse Matrix
 - Matrixgleichungen
 - Zusammenhang mit linearen Gleichungssystemen
 - Rang einer Matrix, Rangkriterium
- Determinanten
 - Definition
 - Zusammenhang mit linearen Gleichungssystemen

Im Rahmen der beschreibenden / deskriptiven Statistik werden folgende Themen behandelt:

- Begrifflichkeiten
- Lage- und Streuungsmaße
- Abhängigkeitsmessung bei qualitativen, komperativen und quantitativen Merkmalen insbesondere Regressionsanalyse
- Deskriptive Zeitreihenanalyse mit Trend-, Saison- und Restkomponentenschätzung nach unterschiedlichen Methoden
- Meß- und Indexzahlen

Lernergebnisse:

Nach der Lehrveranstaltung können die Studierenden ...

- Statistische Daten verdichten und graphisch aussagekräftig darstellen
- Wesentliche Aussagen über Daten anhand geeigneter Kennzahlen treffen und interpretieren
- Die Ableitung von Regressionsformeln verstehen und komplexe Regressions- und deskriptive Zeitreihenanalysen abgestimmt auf den jeweiligen Datensatz durchführen und interpretieren
- sicher im Umgang mit Meß- und Indexzahlen agieren

Nach dem erfolgreichen Besuch der Vorlesung sind die Lernenden in der Lage ...

- lineare algebraische Gleichungssysteme mittels des Gauß-Algorithmus in die Lösbarkeitskategorien (eindeutig lösbar, unendlich viele Lösungen, unlösbar) einzuteilen und ggfs. die Lösung anzugeben.
- die Techniken und Methoden der Vektorrechnung anzuwenden.
- die Techniken und Methoden der Matrixrechnung anzuwenden.
- die Determinante einer niedrigdimensionalen Matrix zu berechnen und den Zusammenhang der Determinante zur Lösungstheorie linearer Gleichungssysteme herzustellen
- einfache technische oder ökonomische Systeme mittels der Techniken und Methoden der linearen Algebra zu modellieren und aus der ermittelten Lösung der mathematischen Formulierung das System quantitativ zu beurteilen.

Voraussetzungen und Empfehlungen:

Grundlegende mathematische Kenntnisse, wie sie im Mathematik-Brückenkurs vermittelt werden, werden vorausgesetzt.

Literatur:

- PAPULA, Lothar:
Mathematik für Ingenieure und Naturwissenschaftler,
Band 2, Teil I. 15. Aufl. Wiesbaden: Springer Vieweg Verlag 2025
- HELM, Werner; PFEIFER, Andreas; OHSER, Joachim:
Mathematik für Wirtschaftswissenschaftler.
1. Aufl. München: Carl Hanser Verlag 2011
- GRAMLICH, Günter:
Lineare Algebra: Eine Einführung.
5. Aufl. München: Carl Hanser Verlag 2021
- TESCHL, Gerald; TESCHL, Susanne:
Mathematik für Informatiker,
Band 1: Diskrete Mathematik und lineare Algebra.
4. Aufl. Heidelberg: Springer Verlag 2013
- FISCHER, Gerd:
Lineare Algebra: Eine Einführung für Studienanfänger.
20. aktualisierte Aufl. Wiesbaden: Springer Verlag 2025
- Christensen, B.; Christensen, S.; Missong, M.: Statistik klipp & klar; 2019; Springer Gabler Verlag
- Bamberg, G.; Baur, F; Krapp, M: Statistik; 18. Auflage; 2017; De Gruyter Oldenbourg Verlag; München
- Missong, Martin; Aufgabensammlung zur deskriptiven Statistik; 2005; 7. Auflage; Verlag R. Oldenbourg, München.
- Schneider, Wolfgang; Kornrumpf, J.; Mohr, Walter; Statistische Methodenlehre --- Definitions- und Formelsammlung zur deskriptiven und induktiven Statistik mit Erläuterungen; 1993; Verlag Oldenbourg, München.

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science

- Betriebswirtschaftslehre Bachelor of Science
 - Computer Games Technology Bachelor of Science
 - Data Science & Artificial Intelligence Bachelor of Science
 - E-Commerce Bachelor of Science
 - Informatik Bachelor of Science
 - IT-Ingenieurwesen Bachelor of Science
 - IT-Management & Consulting Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB032 – Übertragungstechnik

Verantwortliche:	Carsten Burmeister
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB182 – Übertragungstechnik
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	48 Stunden
Aufwand während Semesterferien:	72 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Carsten Burmeister

Studieninhalte:

- Signalformen: Signalanalyse und -synthese
 - Sinusförmige Signale
 - Nicht-sinusförmige periodische Signale
 - Nicht-periodische Signale
- Komplexe Wechselstromrechnung
 - Zeigerdarstellung von sinusförmigen Größen
 - Wechselstromkreise mit Widerstand, Kondensator und Spule
- Ersatzschaltbilder realer elektrischer Bauteile
- Leistungsberechnung im Wechselstromkreis
- Filterstrukturen
 - Tiefpass und Hochpass 1. Ordnung
 - Tiefpass und Hochpass 2. und höherer Ordnung
 - Bandpass und Bandsperre 2. und höherer Ordnung
- Schwingkreise
 - Reihenschwingkreis
 - Parallelschwingkreis
 - Frequenzgänge, Ortskurven und Bodediagramm

Lernergebnisse:

Die Studierenden ...

- besitzen ein Verständnis für Signalformen, -verzerrungen und -verarbeitung bei der Übertragung analoger und diskreter Signale.
- können elektrische Schaltungen in der Nachrichtenübertragung anwenden.
- besitzen die Kenntnis hinsichtlich der Maßnahmen zur Qualitätssicherung bei der Signalübertragung.
- kennen physikalische und logische Übertragungsnetzstrukturen.

Voraussetzungen und Empfehlungen:

- Grundlagen der Elektrotechnik: Sicherer Umgang mit den Grundbegriffen der Elektrotechnik (Strom, Spannung, Widerstand, Leistung) und den grundlegenden Gesetzen (Ohmsches Gesetz,

Kirchhoffsche Regeln).

- Gleichstromnetzwerke: Verständnis für den Aufbau und die Funktionsweise von Gleichstromnetzwerken sowie die Fähigkeit, einfache Schaltungen zu analysieren.

Literatur:

- Hagmann, G.: Grundlagen der Elektrotechnik. Aula-Verlag, 2000 (7. Auflage)
- Meyer, M.: Kommunikationstechnik. Vieweg-Teubner, 2008 (3. Auflage)
- Paul, S.: Grundlagen der Elektrotechnik und Elektronik 2: Elektromagnetische Felder und ihre Anwendungen. Springer-Verlag, 2019 (2. Auflage)
- Kasikci, I.: Wechselstromschaltungen. Springer Vieweg Berlin, Heidelberg, 2025 (5. Auflage)

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB040 – Algorithmen und Datenstrukturen

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB015 – Algorithmen und Datenstrukturen
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	36 Stunden
Aufwand während Semesterferien:	24 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Christian Uhlig

Teilleistung:	TB016 – Übg. Algorithmen und Datenstrukturen
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	75 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	57 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 72 Stunden weicht signifikant von erwarteten 2.0 ECTS ab.</i>
Lehrende:	Malte Heins

Studieninhalte:

- Analyse von Algorithmen
 - Laufzeit und Speicherbedarf
 - Groß-O / Groß-Omega / Groß-Theta Notationen
 - Amortisierte Laufzeitanalyse
 - Iterative vs rekursive Implementierungen
- Sortieren und Suchen
- Listenstrukturen
 - Verkettete Listen (lineare Listen, Ringlisten, einfach und doppelt verkettete Listen)
 - Arraybasierte Listen
 - Skiplisten
- Baumstrukturen
 - Binäre Suchbäume
 - Balancierte Suchbäume: 2-3-Bäume
 - Balancierte Binäre Suchbäume: Rot/Schwarz-Bäume
 - Spreizbäume
 - Tries
 - Arraybasierte Binäre Heaps
- Hash-Tabellen
- Abstrakte Datentypen und ihre Implementierung
 - Listen
 - Mengen
 - Verzeichnisse
 - Warteschlangen
- Java Collections Framework

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden im Rahmen der Übungsaufgaben praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Java behandelt, die nicht Bestandteil der Vorlesung sind.

Lernergebnisse:

Die Studierenden ...

- analysieren, diskutieren und vergleichen einfache Algorithmen und Datenstrukturen hinsichtlich ihres Bedarfs an Laufzeit und Speicher.
- differenzieren bei der Analyse von Algorithmen hinsichtlich best case, worst case und average case.
- differenzieren die Laufzeit von Algorithmen nach ihrem konstanten Faktor und ihrem Wachstum in Abhängigkeit von der Problemgröße.
- beurteilen die Laufzeit von Algorithmen ausgehend von Komplexitätsklassen in den Groß-O-, Groß-Omega- und Groß-Theta-Notationen.
- nennen und erläutern wesentliche Aspekte, Funktionsweisen und Eigenschaften von Algorithmen zum Suchen und Sortieren.
- erläutern die Differenzierung in abstrakte Datentypen und ihre Implementierung.
- nennen und erläutern typische abstrakte Datentypen wie Listen, Mengen, Verzeichnisse und Warteschlangen mit ihren Operationen und Anwendungsbereichen.
- nennen und erläutern Motivation, Funktionsweise und Eigenschaften typischer Implementierungen abstrakter Datentypen mit verketteten Listen, Arrays, Baumstrukturen und Hash-Tabellen.
- wählen zu einer gegebenen Problemstellung einen geeigneten abstrakten Datentypen nebst einer geeigneten Implementierung aus.
- wenden die Elemente allgemein der objektorientierten Programmierung und speziell der Programmiersprache Java zur Lösung algorithmischer Problemstellungen an
- wenden die abstrakten Datentypen und Implementierungen des Java Collections Frameworks an

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden Grundkenntnisse der prozeduralen und der objektorientierten Programmierung, insbesondere in der Programmiersprache Java. Diese Kenntnisse sollten insbesondere die Abbildung abstrakter Datentypen per Interfaces und abstrakter Klassen und die Verwendung einfacher generischer Typen umfassen. Es empfiehlt sich, bereits vorhandenes Grundlagenwissen zu Arraylisten, zu verketteten Listen und zu Sortieralgorithmen im Vorwege aufzufrischen.

Literatur:

- Sedgewick, Robert; Wayne, Kevin: Algorithms, 4th Edition, Addison-Wesley, 2011
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford: Introduction to Algorithms, 3rd Edition, The MIT Press, 2009
- Knuth, Donald E.: The Art of Computer Programming Vol. 1 Fundamental Algorithms, 3rd Edition, Addison-Wesley, 1997
- Knuth, Donald E.: The Art of Computer Programming Vol. 3 Sorting and Searching, 2nd Edition, Addison-Wesley, 1998
- Wirth, Niklaus: Algorithmen und Datenstrukturen, 5. Auflage, Teubner, 2013
- Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D.: The Design and Analysis of Computer Algorithms, 1st Edition, Pearson, 1975
- Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D.: Data Structures and Algorithms, Addison-Wesley, 1983
- Aho, Alfred V.; Ullman, Jeffrey D.: Foundations of computer science, Computer Science Press, 1992
- Dokumentation zur Java-API

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science

- IT-Management & Consulting Bachelor of Science
 - IT-Management / -Consulting & -Auditing Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
-

◆ MB043 – Systemnahe Programmierung

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB072 – Systemnahe Programmierung
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	2,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Christian Uhlig

Teilleistung:	TB074 – Übg. Systemnahe Programmierung
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	35 Min.
ECTS:	3,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	87 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 102 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.</i>
Lehrende:	Malte Heins

Studieninhalte:

- Typische Elemente und Eigenschaften eines C-Programms
- Datentypen
 - Ganzzahl- und Aufzählungstypen, Wahrheitswerte als Ganzzahlen
 - Fließkommatypen, Grundlagen von Fließkommazahlen
 - Strukturierte Typen
 - Vereinigungstypen
 - Zeigertypen
 - Arraytypen
- Funktionszeiger und ihre Anwendungsbereiche
- Konvertierungen
- Arrays und ihre Beziehung zu Zeigern
- Ausdrücke
 - Konstanten
 - Grundlegende Ausdrücke (Zuweisungen, Funktionsaufrufe, etc.)
 - Arithmetische Ausdrücke
 - Boolesche Ausdrücke, Vergleichsoperatoren, logische Operatoren
 - Bitweise Operatoren
 - Arbeit mit Zeigern und Zeigerarithmetik
 - Vorrang und Assoziativität
 - Aspekte der Auswertung (Auswertungsreihenfolge, verkürzte Auswertung, sequence points)
- Anweisungen, insbesondere Verzweigungen und Schleifen
- Dynamische Speicherverwaltung
- Übersetzungsprozess und C-Präprozessor
- Funktionsaufrufe in Maschinen, Aufrufstapel
- Gefahren der Sprache C am Beispiel eines Buffer Overflows mit Manipulation der Rücksprungadresse

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden im Rahmen der Übungsaufgaben praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache C und der C-Standardbibliothek behandelt, die nicht Bestandteil der Vorlesung sind.

Lernergebnisse:

Die Studierenden ...

- formulieren Programme in der Programmiersprache C unter Berücksichtigung der Besonderheiten der Programmiersprache insbesondere in Hinblick auf undefiniertes Verhalten, Plattformabhängigkeiten und Unsicherheiten bestimmter Sprachkonstrukte (z.B. Zeigerarithmetik und fehlende Boundary Checks).
- erläutern in groben Zügen die Repräsentation von Daten und die Abläufe in einem Rechner bei der Ausführung von Anweisungen und Auswertung von Ausdrücken in einer höheren Programmiersprache, insbesondere im Rahmen von Unterprogrammaufrufen.
- erstellen maschinennahe Programme unter besonderer Berücksichtigung von Effizienzaspekten bezogen auf den konstanten Faktor des realisierten Algorithmus
- erläutern typische Gefahren bei Verwendung der Programmiersprache C wie z.B. buffer overflows und berücksichtigen diese Aspekte in der Softwareentwicklung

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden Grundkenntnisse in statisch getypten imperativen Programmiersprachen, die insbesondere charakteristische Datentypen und Kontrollstrukturen (Sequenz, Selektion, Iteration) umfassen und idealerweise auch bereits den Umgang mit Zeigern. Diese Kenntnisse sollten mit gefestigter Programmierpraxis in einer entsprechenden Sprache verbunden sein. Gegebenenfalls empfiehlt es sich, die Kenntnisse am Beispiel einfacher Programmieraufgaben im Vorwege aufzufrischen, um den Einstieg zu erleichtern.

Weiterhin wird ein sicherer Umgang mit der Kommandozeile zum Einsatz der Softwarewerkzeuge in der Übung vorausgesetzt. Gegebenenfalls empfiehlt es sich, die entsprechenden Kenntnisse vorzugsweise am Beispiel der UNIX-Kommandozeile im Vorwege aufzufrischen.

Literatur:

- Harbison, Samuel; Steele, Guy L.: C - A Reference Manual, 5th edition, Prentice Hall, New Jersey, 2002
- Kernighan, Brian W.; Ritchie, Dennis M.: C Programming Language, Prentice Hall, New Jersey, 1998
- Standard zur Programmiersprache, insbesondere ISO/IEC 9899:1999 und ISO/IEC 9899:2011

Studiengänge:

- Computer Games Technology Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB048 – Elektronik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB185 – Elektronik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	5.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	2 Stunden
Flexibel einteilbarer Aufwand:	3 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 60 Stunden weicht signifikant von erwarteten 5.0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Passive Bauelemente und Schalter
 - Übersicht der elektronischen Bauelemente, E-Normreihen
 - Widerstand, Kondensator, Spule
 - Schalter
- Thermisches Verhalten von Bauelementen
 - Einleitung und Grundbegriffe, Wärmestromkreis
 - Aspekte des thermischen Verhaltens
 - Erwärmung und Abkühlung, Einsatz von Kühlkörpern
- Lineare Netzwerke bei Gleichstrom
 - Strom- und Spannungsquellen, Grundstromkreis
 - Bestimmung des Arbeitspunktes
 - Allgemeine Netzwerkanalyse
- Lineare Netzwerke bei zeitabhängiger Erregung
 - Wechselstrom und Wechselspannung
 - Passive Filter, Schwingkreise und Resonatoren
 - Wechselstrombrücken
 - Ausgleichsvorgänge
- Anwendungen der Systemanalyse
 - Einleitung
 - Fourier-Reihen, Klirrfaktor
 - Fourier- und Laplace-Transformation
 - Vierpoltheorie

Lernergebnisse:

Die Studierenden ...

- kennen die theoretischen Grundlagen der Elektronik
- kennen Methoden und Werkzeuge für Entwurf und Analyse von elektronischen Systemen
- können das reale Verhalten von elektronischen Bauteilen (Toleranzen, Temperaturabhängigkeiten usw.) beim Schaltungsentwurf berücksichtigen
- können einfache analoge und digitale Systeme aus dem Bereich der technischen Informatik begreifen, spezifizieren und entwerfen

- können mit modernen Entwurfs- und Simulationswerkzeugen arbeiten

Voraussetzungen und Empfehlungen:

Kenntnisse der elektrotechnischen Grundlagen, Umgang mit komplexen Zahlen und Funktionen, Infinitesimalrechnung

Literatur:

- Tietze, Ulrich; Schenk, Christoph: Halbleiterschaltungstechnik, 16. Auflage Springer Verlag, 2016
- Stiny, Leonhard: Handbuch passiver elektronischer Bauelemente, Franzis Verlag 2007
- Hering, Ekbert; Bessler, Klaus; Gutekunst, Jürgen: Elektronik für Ingenieure und Naturwissenschaftler, Springer Verlag 2005
- Lunze, Klaus: Berechnung elektrischer Stromkreise, 15. Auflage Huss Medi 1990
- Horowitz, Paul; Hill, Winfield: Die Hohe Schule der Elektronik. Teil 1: Analogtechnik, 8. Auflage Elektor-Verlag 2006
- Schiffmann, Wolram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Brauer, Harry; Lehmann, Constans; Lindner, Helmut: Taschenbuch der Elektrotechnik und Elektronik, 9. Auflage Hanser Fachbuchverlag, 2008

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB173 – Projekt Eingebettete Software

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB209 – Projekt Eingebettete Software
Lernform:	Projektarbeit
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	15 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	50 Stunden
Aufwand während Semesterferien:	25 Stunden
Flexibel einteilbarer Aufwand:	60 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

themenabhängig

Lernergebnisse:

Die Studierenden...

- kennen wesentliche Programmstrukturen und Mechanismen, die in eingebetteten Computern als Basis smarter Systeme zum Tragen kommen.
- kennen die verschiedenen Arten der Dokumentation smarter Systeme.
- können die Struktur der Dokumentation smarter Systeme benennen und können einfache Dokumente anfertigen.
- haben die Fähigkeit einfache Fehlerdiagnosen bei eingebetteten Computern strukturiert durchzuführen.

Voraussetzungen und Empfehlungen:

Für eine erfolgreiche Teilnahme werden grundlegende Kenntnisse in der Programmierung sowie ein erstes Verständnis von Mikrocontrollern und eingebetteten Systemen vorausgesetzt.

Literatur:

themenabhängig

Studiengänge:

- Smart Technology Bachelor of Science

◆ MB375 – Betriebswirtschaftslehre

Verantwortliche:	Fikret Koyuncu
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB371 – Betriebswirtschaftslehre
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Fikret Koyuncu

Studieninhalte:

Die Studierenden erlernen Grundtatbestände der Betriebswirtschaftslehre, beginnend vom Erfahrungs- und Erkenntnisobjekt dieser wissenschaftlichen Disziplin, über zu fällende konstitutive Entscheidungen, bis hin zu den diversen betriebswirtschaftlichen Funktionen innerhalb eines Betriebes.

Letztere stehen im Mittelpunkt der Veranstaltung. Die theoretischen Inhalte werden durch Praxisbeispiele untersetzt.

Durch zahlreiche Übungen wird das Verständnis für die betriebswirtschaftlichen Prozesse und deren Zusammenhänge gefestigt sowie das eigenständige Arbeiten gefördert.

Inhalte der Veranstaltung sind im Einzelnen:

- Betriebswirtschaftslehre als wissenschaftliche Disziplin
- Grundlagen der Betriebswirtschaftslehre
- Rechnungswesen
- Unternehmensführung
- Materialwirtschaft
- Produktionswirtschaft
- Marketing & Absatz
- Investition & Finanzierung
- Umfangreiche Übungen zu verschiedenen Vorlesungsteilen

Lernergebnisse:

Die Studierenden können ...

- das Erfahrungs- und Erkenntnisobjekt der Betriebswirtschaftslehre benennen,
- die Begriffe Wirtschaften und Ökonomisches Prinzip erklären sowie eine Break-Even-Analyse durchführen,
- Unternehmensziele aufzählen; die Aufgaben der Zielbildung erläutern sowie den Zielbildungsprozess wiedergeben,
- ausgewählte Kennzahlen ausrechnen,
- Ziele der Unternehmensführung erläutern, Führungsebenen voneinander abgrenzen, den Führungsprozess beschreiben sowie ausgewählte Führungsstile erläutern und -prinzipien erklären,
- Ableiten des Begriffsinhalts, der Bedeutung, der Funktion und der Teilgebiete des Rechnungswesens,
- Durchführen der buchhalterischen Erfassung ausgewählter Geschäftsvorfälle,

- Einführung, Begriffserklärungen, Kostentheorie,
- Betriebsabrechnungsbogen und Preiskalkulation
- die Ziele der Materialwirtschaft wiedergeben und durch Anwendung von Methoden materialwirtschaftliche Analysen durchführen und Handlungsanweisungen ableiten,
- ausgewählte Erzeugnisstrukturdarstellungen für gegebene Problemstellungen erstellen und mit programmorientierten Verfahren die Materialbedarfsplanung durchführen,
- mit ausgewählten Verfahren die optimale Bestellmenge bestimmen,
- den Input, Throughput und Output von Produktionsprozessen beschreiben,
- das optimale Produktionsprogramm für ausgewählte Fälle ermitteln,
- ausgewählte Aufgaben der Produktionsprozessplanung ausführen,
- die Ziele des Marketings nennen, Methoden zur Ableitung der Marketing-Strategie beschreiben und anwenden sowie die Instrumente des Marketing-Mix erläutern,
- Investitionsarten voneinander abgrenzen; den Investitionsprozess beschreiben und die Aufgabe der Investitionskontrolle skizzieren sowie die Vorteilhaftigkeit einer Investition mittels Methoden beurteilen,
- die Ziele und Aufgaben der Finanzwirtschaft nennen; die Finanzierung aus Abschreibungen erläutern sowie den Financial-Leverage-Effekt an einem Beispiel demonstrieren,
- die Bedeutung informationstechnischer Systeme zur Bewältigung betriebswirtschaftlicher Aufgaben erläutern.

Voraussetzungen und Empfehlungen:

keine

Literatur:

- BECKER, Hans Paul: Investition und Finanzierung. 7. akt. Aufl. Wiesbaden: Gabler, 2016
- BERNECKER, Michael: Grundlagen der Betriebswirtschaftslehre. 4. Aufl. Köln: Johanna, 2011.
- BLOHM, Hans; LÜDER, Klaus; SCHÄFER, Christina: Investition. 10. akt. Aufl. München: Vahlen, 2012
- DÄUMLER, Klaus-Dieter; GRABE, Jürgen: Grundlagen der Investitions- und Wirtschaftlichkeitsrechnung. 12. vollst. überarbeitete Aufl. Berlin; Herne: Neue Wirtschafts-Briefe, 2007
- JUNG, Hans: Allgemeine Betriebswirtschaftslehre. akt. 13. Aufl. München: Oldenbourg, 2016
- SCHIERENBECK, Henner; WÖHLE, Claudia: Grundzüge der Betriebswirtschaftslehre. 18. überarb. Aufl. München: Oldenbourg, 2012
- SPECHT, Olaf; SCHMITT, Ulrich: Betriebswirtschaftslehre für Ingenieure + Informatiker. 5. Aufl. München; Wien: Oldenbourg, 2000
- THOMMEN, Jean-Paul; ACHLEITNER, Ann-Kristin: Allgemeine Betriebswirtschaftslehre. 7. vollst. überarb. Aufl. Wiesbaden: Gabler, 2012
- VAHS, Dietmar; SCHÄFER-KUNZ, Jan: Einführung in die Betriebswirtschaftslehre. 7. überarb. Aufl. Stuttgart: Schäffer-Poeschel, 2015.
- WEBER, Wolfgang; KABST, Rüdiger: Einführung in die Betriebswirtschaftslehre. 9. akt. u. überarb. Aufl. Wiesbaden: Gabler, 2014
- WÖHE, Günter; DÖRING, Ulrich: Einführung in die Allgemeine Betriebswirtschaftslehre. 26. überarbeitete und aktualisierte Aufl. München: Vahlen, 2016

Studiengänge:

- Computer Games Technology Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB057 – Fortgeschrittene Objektorientierte Programmierung

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB024 – Fortgeschrittene Objektorientierte Programmierung
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Christian Uhlig

Teilleistung:	TB025 – Übg. Fortgeschrittene Objektorientierte Programmierung
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	75 Min.
ECTS:	3.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	86 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 101 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Malte Heins

Studieninhalte:

- Generische Typen / Java Generics
- Verschachtelte Typen
- Funktionswerte in OO-Sprachen (Funktionale Interfaces, Lambda-Ausdrücke, Methodenreferenzen)
- Funktionale Programmierung mit Java Streams
- Nebenläufigkeit (Threads, Racing Conditions, Synchronisation, Waitsets, volatile Variablen, Java Memory Model / happens-before-Relation, nebenläufige Auswertung von Streams, Executors, Futures, Fork-Join-Tasks, Thread-sichere Collections)

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden im Rahmen der Übungsaufgaben praxisrelevante Aspekte der fortgeschrittenen OOP behandelt, die nicht Bestandteil der Vorlesung sind.

Lernergebnisse:

Die Studierenden ...

- wenden fortgeschrittene Konzepte objektorientierter Programmiersprachen wie parametrische Polymorphie und Funktionswerte zur Konstruktion wiederverwendbarer Softwarekomponenten an.
- nennen und erläutern die Grundlagen generischer Datentypen und ihre Korrespondenz mit Konzepten objektorientierter Sprachen (Schnittstellen, abstrakte Klassen, konkrete Klassen, Polymorphie).
- entwerfen und implementieren generische Datentypen mit Java Generics und nutzen hierbei beschränkte Typparameter, generische Methoden und parametrisierte Typen mit Wildcards.

- definieren Funktionswerte per anonymer Klassen, Lambda-Ausdrücke und Methodenreferenzen.
- wenden Prinzipien funktionaler Programmierung in objektorientierten Sprachen am Beispiel von Java Streams an.
- nennen und erläutern Motivation, Grundlagen und Herausforderungen nebenläufiger Programmierung.
- wenden Primitiven nebenläufiger Programmierung in Java an (Erzeugen von Threads, Thread-Kommunikation / Synchronisation, usw.).
- nennen und erläutern die Herausforderungen gemeinsamer Zustände von Threads, das Java-Speichermodell und die happens-before-Relation und wenden diese Kenntnisse zur Vermeidung von racing conditions und data races an.
- nennen und erläutern die Vorzüge funktionaler Programmierung für die parallele Auswertung von Berechnungen am Beispiel von Java Streams.
- definieren Stream-Pipelines unter Berücksichtigung der Anforderungen und Konsequenzen einer nebenläufigen Auswertungsstrategie.

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden gefestigte theoretische und praktische Kenntnisse in objektorientierter Programmierung im Allgemeinen und in der Programmiersprache Java im Speziellen. Gegebenenfalls empfiehlt es sich, die Kenntnisse im Vorwege anhand einfacher Übungsaufgaben aufzufrischen.

Literatur:

- Gosling, James; Joy, Bill; Steele, Guy; Bracha, Gilad; Buckley, Alex; Smith, Daniel: The Java Language Specification, Java SE 11 Edition, 2018
- Bloch, Joshua: Effective Java: A Programming Language Guide, 3rd Edition, Addison-Wesley, 2017
- Lea, Doug: Concurrent Programming in Java: Design Principles and Patterns, 2nd Edition, Addison-Wesley, 1999
- Goetz, Brian; Peierls, Tim; Bloch, Joshua; Bowbeer, Joseph; Holmes, David; Lea, Doug: Java Concurrency in Practice, Addison-Wesley, 2006
- Urma, Raoul-Gabriel; Fusco, Mario; Mycroft, Alan: Java 8 in Action: Lambdas, Streams, and Functional-Style Programming, Manning Publications, 2014
- Ullenboom, Christian: Java ist auch eine Insel, 12. Auflage, Galileo Press GmbH, 2016
- Loudon, Kenneth C.: Programming Languages: Principles and Practice, 2nd Edition, Thomson Learning, 2002

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB059 – Web-Anwendungen

Verantwortliche:	Marian Gajda
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB027 – Web-Anwendungen
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	3 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	22 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	82 Stunden
Lehrende:	Marian Gajda

Teilleistung:	TB028 – Übg. Web-Anwendungen
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	45 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	40 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	55 Stunden
Lehrende:	Marian Gajda

Studieninhalte:

- Basiskonzepte des WWW
 - Klassische Auszeichnungsmöglichkeiten in HTML
 - HTML-Formulare und ihre Möglichkeiten
 - Style Sheets
 - CSS-Animationen
 - Templating
 - Responsive Design
- Dynamik in Web-Seiten mit Javascript
 - Client-seitige Dynamik
 - Server-seitige Dynamik
- Asynchronous Javascript

Bearbeitung von Übungsaufgaben, die sich am Stoff der Vorlesung orientieren, in Zweiergruppen mit Abnahme der Lösungen. Erstellt wird eine im Verlaufe der einzelnen Übungseinheiten komplexer werdende Web-Anwendung, wobei die einzelnen Schritte aufeinander aufbauen, so dass am Ende eine komplexe Web-Anwendung entsteht, die einen Großteil der in der Vorlesung erlernten Techniken und Konzepte nutzt.

Lernergebnisse:

Die Studierenden ...

- sind in der Lage, die in der Vorlesung vermittelten theoretischen Hintergründe selbst praktisch anzuwenden.
- haben umfangreiche Kenntnisse und praktische Erfahrungen zu den Themen HTML, CSS, serverseitiger Dynamik, clientseitiger Dynamik mit JavaScript und AJAX, Einsatz JSON zum Austausch von Daten zwischen Client und Server, Einsatz von Cookies und Sessions zum temporären Speichern von Daten.
- steigern ihre Teamfähigkeit durch intensive Arbeit in Zweierteams und Kommunikation über auftretende Probleme in der ganzen Gruppe.

Die Studierenden ...

- führen die technischen Randbedingungen des Internet auf und benennen ihre Auswirkungen.
- beschreiben die konzeptionellen Aspekte von Stylesheets und der zentralen Möglichkeiten zur Festlegung der Darstellung in den Cascading Stylesheets und nutzen diese zur Erzeugung angestrebter Darstellungsweisen.
- können responsive Web-Layouts erstellen
- kennen wichtige Konzepte, Sprachen, Frameworks und Architekturen zur Realisierung dynamischer Webseiten auf, wählen zwischen diesen problembezogen aus und nutzen sie zur Erstellung dynamischer Webseiten.
- geben die zusätzliche Konzepte und Sprachelemente von HTML 5 an und entwerfen damit Webseiten.
- nutzen die theoretisch vermittelten Inhalte zur eigenständigen Realisierung von Webanwendungen begrenzter Komplexität.

Voraussetzungen und Empfehlungen:

Die Studierenden müssen gegebenen Quelltext lesen und fehlerfrei reproduzieren können. Die in vorigen Veranstaltungen erlernten Fertigkeiten im Umgang mit einem Versionskontrollsystem wird in der Übung vorausgesetzt. Es ist hilfreich, wenn die Grundlagen der Netzwerktechnik (Hostname, IP-Adresse, Port) bekannt sind.

Zur Vorbereitung auf Heimarbeit kann ein beliebiger Quelltext-Editor installiert werden.

Literatur:

- WOLF, Jürgen: HTML5 und CSS - Das umfassende Handbuch 2019
- ACKERMANN, Philipp: JavaScript - Das umfassende Handbuch 2019
- LABORENZ, Kai: CSS: Das umfassende Handbuch. Galileo Computing, 2011
- GASSTON, Peter: Moderne Webentwicklung: Geräteunabhängige Entwicklung - Techniken und Trends in HTML5, CSS3 und JavaScript, dpunkt.verlag, 2014
- WORLD WIDE WEB CONSORTIUM: HTML 5. <http://www.w3.org/TR/2014/WD-html5-20140617/>

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB068 – Halbleiterschaltungstechnik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB186 – Halbleiterschaltungstechnik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	65 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	5 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Sergei Sawitzki

Teilleistung:	TB190 – Übg. Elektronik und Halbleiterschaltungstechnik
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	5 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Timm Bostelmann

Studieninhalte:

- Grundlagen der Halbleiterphysik
- Halbleiter-Bauelemente
 - pn-Übergang, Diode
 - Bipolartransistor
 - Feldeffekttransistoren
 - Überblick über sonstige Bauelemente
 - Grundsaltungen, Verstärker
- Halbleiterschaltungstechnik
 - Strom- und Spannungsquellen
 - Operationsverstärker
 - Transistoren als Schalter, Digitale Schaltungen
- Passive Netze
 - Filterschaltungen
 - Einschwingvorgänge
 - Stabilisierungsschaltungen
- Verstärkerschaltungen
 - Arbeitspunkteinstellungen
 - Statisches und dynamisches Verhalten
 - Gegenkopplungen
- Operationsverstärkerschaltungen
 - Beschaltungen
 - Anwendungen
 - Stabilität

Lernergebnisse:

Die Studierenden ...

- können analoge und digitale elektronische Systeme, soweit sie für die technische Informatik von Belang sind, verstehen und entwerfen
- kennen die Methodik des Schaltungsentwurfs und können diese praktisch einsetzen
- haben ein Verständnis für realitätsnahe Schaltungsentwicklung unter Einbeziehung des realen Bauteilverhaltens (Toleranzen, Streuungen, Temperaturabhängigkeiten usw.)
- können bestehende Schaltungen und Systeme analysieren

- kennen den Aufbau und die Funktionsweise von den wichtigsten Halbleiter-Bauelementen
- erkennen Verbindungen zwischen Digitaltechnik und Halbleiterschaltungstechnik als technologischer Grundlage digitaler Schaltungen und Systeme
- können typische Aufgaben aus dem Stoffumfang der Vorlesungen Elektronik und Halbleiterschaltungstechnik eigenständig lösen
- Kenndaten elektronischer Bauteile aus Datenblättern entnehmen

Voraussetzungen und Empfehlungen:

Erfahrungen im Umgang mit Simulationswerkzeugen (insbesondere Pspice)

Literatur:

- Tietze, Ulrich; Schenk, Christoph: Halbleiterschaltungstechnik, 16. Auflage Springer Verlag, 2016
- Stiny, Leonhard: Handbuch aktiver elektronischer Bauelemente, Franzis Verlag 2009
- Hering, Ekbert; Bessler, Klaus; Gutekunst, Jürgen: Elektronik für Ingenieure und Naturwissenschaftler, Springer Verlag 2005
- Horowitz, Paul; Hill, Winfield: Die Hohe Schule der Elektronik. Teil 1: Analogtechnik, 8. Auflage Elektor-Verlag 2006
- Schiffmann, Wolfram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Brauer, Harry; Lehmann, Constans; Lindner, Helmut: Taschenbuch der Elektrotechnik und Elektronik, 9. Auflage Hanser Fachbuchverlag 2008
- Taur, Yuan; Ning, Tak H.: Fundamentals of Modern VLSI Devices, 2nd edition, Cambridge University Press 2009
- Ng, Kwok K.: Complete Guide to Semiconductor Devices, 2nd edition, Jaohn Wiley & Sons 2002

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB097 – Bildbearbeitung und -analyse

Verantwortliche:	Dennis Säring
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB076 – Bildbearbeitung und -analyse
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Dennis Säring

Teilleistung:	TB083 – Prakt. Bildbearbeitung und -analyse
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	16 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	91 Stunden
Lehrende:	Hermann Höhne

Studieninhalte:

- Einführung in die Bildbearbeitung
- Visualisierung und Bildanpassung
- Komposition und Filterung
- Fourier-Transformation und Frequenzfilter
- Lineare und nicht-lineare Registrierung
- Segmentierung und Texturanalyse
- Klassifikationsverfahren
- Selbstständiges Einarbeiten in den C++17 Standard auf Basis des vorhandenen Hintergrundwissens um C
- Selbstständiges Einarbeiten in die Basisfunktionen von OpenCV bei minimaler Hilfestellung
- Implementierung von Algorithmen zur Bildbearbeitung und -analyse:
 - Punktweise Transformationen (Fensterung, Gamma-Korrektur)
 - Lineare Faltungsfilter und morphologische Filter
 - Affine Transformationen und Interpolation
 - Farb- und Texturanalyse
 - Grundlagen der Objekterkennung inklusive Segmentierung und Formerkennung

Lernergebnisse:

Die Studierenden ...

- haben ein Verständnis für die Struktur digitale Bilddaten
- erkennen die Möglichkeiten der Darstellung und Anpassung von digitalen Bildern
- können das Konzept der Bildkomposition und der Bildfilterung in praktischen Übungen umsetzen
- sind in der Lage Bilddaten in Frequenz- und Ortsbereich zu analysieren und zu bearbeiten
- verfügen über die theoretischen Grundlagen zur Registrierung von Bilddaten
- kennen unterschiedliche Ansätze Objekte in Bilddaten automatisch zu klassifizieren und zu segmentieren
- sammeln im Praktikum Erfahrungen beim Implementieren von Algorithmen zur Bildbearbeitung
- vertiefen durch praktisches Umsetzen die in der zugehörigen Vorlesung theoretisch erläuterten Algorithmen zur Bildbearbeitung und -analyse
- üben das selbstständige Erweitern ihrer Kenntnisse in Programmiersprachen
- üben das selbstständige Einarbeiten in eine vorgegebene Bibliothek

- können diese Fähigkeiten im kommerziellen oder akademischen Umfeld praktisch nutzen

Voraussetzungen und Empfehlungen:

Zur Vorbereitung auf das Praktikum kann auf dem heimischen Rechner eine Entwicklungsumgebung für C++ installiert werden. Das in den vorigen Veranstaltungen erworbene Wissen über die Syntax von C und die objektorientierten Konzepte von Java können zwecks Kombination wiederholt werden.

Literatur:

- Burger, Burge: Digitale Bildverarbeitung, Springer-Verlag 2005
- Handels: Medizinische Bildverarbeitung, Vieweg+Teubner 2009
- Howse, Minichino: Learning OpenCV 4 Computer Vision with Python 3, Packt Publishing 2020

Studiengänge:

- Computer Games Technology Bachelor of Science
- Informatik Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB118 – Soft Skills

Verantwortliche:	Jan-Paul Lüdtkke
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB042 – Assistenz
Lernform:	Assistenz
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	
ECTS:	3,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	3 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	22 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	68 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Jan-Paul Lüdtkke

Teilleistung:	TB043 – Communication Skills
Lernform:	Workshop
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	
ECTS:	2,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Anna-Magdalena Kölzer

Studieninhalte:

Communication Skills

Im Rahmen des Workshops werden die folgenden Inhalte behandelt:

- Selbstanalyse (Fragebogen) als Grundlage für Bewerbungen
- Rhetorik & Präsentation (Theorie und Praxis)
- Struktur und Aufbau von Bewerbungsunterlagen
- Bewerbungsprozess
- Interview (Theorie und Praxis)
- Assessment Center (Theorie)
- Persönlichkeitsfragebogen und Testverfahren (Intelligenz und Konzentration) (Praxis)
- Gruppenübungen (Praxis)

Assistenz

Im Rahmen der Assistenz werden die Studierenden von den Hochschullehrenden mit konkreten (Teil-)Projekten betraut. Diese können ein weites Spektrum umfassen. So sind z.B. die Durchführung kleinerer empirischer Umfragen oder auch die eigenständige Recherche und Ausarbeitung spezieller Fachinhalte denkbar. Ebenso in Betracht kommen die Durchführung von Tutorien oder Übungen. Die Assistenz ist selbstständig zu bearbeiten und kann die Abstimmung mit anderen Studierenden erfordern.

Lernergebnisse:

Nach Abschluss des Moduls haben die Studierenden die Fähigkeit erworben, in Kooperation mit den Dozenten und Assistenten, ihr Wissen und ihre Erfahrungen aus früheren Veranstaltungen der Betriebswirtschaftslehre, Mathematik und Informatik an Studierende jüngerer Semester weiter zu geben. Mit zunehmender Dauer des Semesters verbinden die Studierenden Kenntnisse aus dem Workshop "Communication Skills" mit ihrer Assistenz Tätigkeit.

Die Studierenden verfügen nach dem Besuch des Workshops über folgende Kompetenzen:

- Besitz verbesserter persönlicher Soft Skills, wie sie für Studium oder Beruf erforderlich sind
- Sensibilität für menschliche Interaktionen und Betriebsprozesse
- Besitz erweiterter rhetorischer Fähigkeiten im Rahmen von Präsentationen, Vorträgen und Referaten sowie sozialer Kompetenz

- Kenntnis der Bedeutung von verbalen und nonverbalen Signalen für die eigene Kommunikation sowie die Fähigkeit, diese zu erkennen
- Fähigkeit zum angemessenen Verhalten bei Teamarbeit oder Projekten
- Fähigkeit zur Selbstdarstellung bei Bewerbungen, Interviews, Assessment-Centern.

Die Studierenden entwickeln im Rahmen der Assistenz unter Anleitung eines Hochschullehrers die Fähigkeiten ...

- fachspezifische Aufgabenstellungen zu analysieren
- problemspezifische Lösungen zu konzipieren und
- als Ergebnis begründet zu präsentieren.

Voraussetzungen und Empfehlungen:

Fachliche Inhalte der ersten vier Studiensemester

Literatur:

- ARNOLD, Frank:
Management von den besten lernen.
München: Hans Hauser Verlag, 2010
- APPELMANN, Björn:
Führen mit emotionaler Intelligenz.
Bielefeld: Bertelsmann Verlag, 2009
- BIERKENBIEHL, Vera F.:
Rhetorik, Redetraining für jeden Anlass. Besser reden, verhandeln, diskutieren.
12. Aufl. München: Ariston Verlag, 2010
- BOLLES, Nelson:
Durchstarten zum Traumjob. Das ultimative Handbuch für Ein-, Um- und Aufsteiger.
2. Aufl. Frankfurt/New York: Campus Verlag, 2009
- DUDENREDAKTION mit HUTH, Siegfried A.:
Reden halten - leicht gemacht. Ein Ratgeber.
Mannheim/Leipzig: Dudenverlag, 2007
- GRÜNING; Carolin; MIELKE; Gregor:
Präsentieren und Überzeugen. Das Kienbaum Trainingskonzept.
Freiburg: Haufe-Lexware Verlag, 2004
- HERTEL, Anita von:
Professionelle Konfliktlösung. Führen mit Mediationskompetenz.
Handelsblatt, Bd., 6, Kompetent managen.
Frankfurt: Campus Verlag, 2009
- HESSE, Jürgen; SCHRADER, Hans Christian:
Assessment-Center für Hochschulabsolventen.
5. Auflage, Eichborn: Eichborn Verlag, 2009
- MENTZEL, Wolfgang; GROTZFELD, Svenja; HAUB, Christine:
Mitarbeitergespräche.
Freiburg: Haufe-Lexware Verlag, 2009
- MORITZ, Andr; RIMBACH, Felix:
Soft Skills für Young Professional. Alles was Sie für ihre Karriere wissen müssen.
2. Aufl. Offenbach: Gabal Verlag, 2008
- PERTL, Klaus N.:
Karrierefaktor Selbstmanagement. So erreichen Sie ihre Ziele.
Freiburg: Haufe-Verlag, 2005
- PORTNER, Jutta:
Besser verhandeln. Das Trainingsbuch.
Offenbach: Gabal Verlag, 2010

- PÜTTJER, Christian; SCHNIERDA, Uwe:
Assessment-Center. Training für Führungskräfte.
Frankfurt/New York: Campus Verlag, 2009
- PÜTTJER, Christian; SCHNIERDA, Uwe:
Das große Bewerbungshandbuch.
Frankfurt: Campus Verlag, 2010
- SCHULZ VON THUN, Friedemann; RUPPEL, Johannes; STRATMANN, Roswitha:
Miteinander Reden. Kommunikationspsychologie für Führungskräfte.
10. Auflage, Reinbek bei Hamburg: rororo, 2003

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB122 – IT-Sicherheit

Verantwortliche:	Gerd Beuster
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB048 – IT-Sicherheit
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Gerd Beuster

Studieninhalte:

- Gegenstandsbereich der IT-Sicherheit
- Aktuelle Richtlinien, Standards, Normen und Gesetze
- Bedrohungen der IT-Sicherheit und daraus resultierende Risiken
- Primäre Sicherheitsziele
- Überblick über Verfahren zur Erreichung der Ziele
- Kryptografische Verfahren
 - Verschlüsselungsverfahren
 - Symmetrische Verschlüsselungsverfahren
 - Asymmetrische Verschlüsselungsverfahren
 - Hash-Funktionen
 - Schlüsselmanagement
 - Zertifikate
 - Kryptografische Protokolle
 - Digitale Signatur
 - Zeitstempel
 - TLS-Protokoll
- Authentifizierungsverfahren
- Übertragungssicherheit in Netzen
 - Sichere IP-Kommunikation
 - VPN-Technologien
- Sicherheitsarchitekturen und ihre Komponenten
 - Sicherheitsaspekte von Web-Servern
 - Firewall-Systeme
 - Intrusion Detection-Systeme
- Sicherheit von Web-Anwendungen
- Sicherheit bei Nutzung von Large Language Models (LLMs)
- Technisch / organisatorische Maßnahmen zur Erhöhung der IT-Sicherheit
- Risiko- und Sicherheitsmanagement

Lernergebnisse:

Die Studierenden erwerben die notwendigen Kenntnisse, um Softwaresysteme und ihre betrieblichen Einsatzszenarien in Hinblick auf ihre Sicherheit einschätzen zu können. Sie sind in der Lage, bei der

Konzeption und Entwicklung von Softwaresystemen und in ihrem Unternehmenseinsatz relevante Sicherheitsaspekte zu berücksichtigen.

- Kenntnis der unterschiedlichen Bedrohungsszenarien und -arten.
- Kenntnis der besonderen Gefahren bei internetbasierten Anwendungen.
- Kenntnis typischer primärer Sicherheitsziele (Vertraulichkeit, Authentifizierung, Verbindlichkeit, u.a.).
- Kenntnis der Verfahren zur Gewährleistung der unterschiedlichen Sicherheitsziele.
- Kenntnis der praxisrelevanten kryptografischen Verfahren und Protokolle.
- Kenntnis der Sicherungsmaßnahmen in Rechnernetzen.
- Fähigkeit, grundlegende Sicherungsmaßnahmen für Web-Anwendungen umzusetzen.
- Fähigkeit, Large Language Models (LLMs) sicher in Software-Systeme einzubinden
- Fähigkeit, die Qualität von durch LLMs generiertem Code einzuschätzen
- Kenntnis der Bestandteile einer IT-Sicherheitsinfrastruktur und ihrer zentralen Funktionalitäten.
- Kenntnis der Verfahren zur Risikoabschätzung und Bewertung der Sicherheit von IT-Systemen und die Fähigkeit, diese anzuwenden.

Voraussetzungen und Empfehlungen:

Das Modul setzt grundlegende Kenntnisse der Programmierung und des Aufbaus eines Computersystems sowie von Computernetzen voraus.

Literatur:

- Anderson, Ross J.: Security Engineering : A Guide to Building Dependable Distributed Systems. 3. Auflage. Hoboken (NJ), USA: Wiley & Sons, 2020.
- BSI - Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): Informationssicherheit und IT-Grundschutz : BSI-Standards 200-1, 200-2 und 200-3. 1. Auflage. Bonn : BSI, 2017.
- Eckert, Claudia: IT-Sicherheit : Konzepte - Verfahren - Protokolle. 11. Auflage. München : Oldenbourg, 2023.
- Ferguson, Niels; Schneier Bruce, Kohno; Tadayoshi: Cryptography Engineering : Design Principles and Practical Applications. Hoboken (NJ), USA: Wiley & Sons, 2010.
- Kersten, Heinrich; Klett, Gerhard: Der IT Security Manager. 4. Auflage. Wiesbaden: Springer Vieweg, 2015.
- Pfleeger, Charls P.;Pfleeger, Shari Lawrence: Security in Computing. 6. Auflage. München: Prentice Hall, 2024.
- Progutke, Werner: Basiswissen IT-Sicherheit : Das Wichtigste für den Schutz von Systemen & Daten. 3. Auflage. Heidelberg: Springer Campus, 2017.
- Stallings, William: Computer Security : Principles and Practice. 5. Auflage. München: Pearson, 2024.
- Stallings, William: Cryptography and Network Security : Principles and Practice. 8. Auflage. München: Pearson, 2023.
- Swoboda, Joachim; Spitz, Stephan; Pramateftakis, Michael: Kryptographie und IT-Sicherheit : Grundlagen und Anwendungen. 2. Auflage Wiesbaden: Vieweg + Teubner Verlag, 2011.

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science

- Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
-

◆ MB168 – Workshop Rapid Manufacturing

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB143 – Workshop Rapid Manufacturing
Lernform:	Workshop
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	15 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	70 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	30 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

- Vorstellung FabLab verfügbarer Anlagen für Rapid Manufacturing
- Erarbeitung und Beschreibung eigener Werkteile anhand ausgewählten Aufgabestellungen aus dem Bereich Smarter Geräte
- Praktischer Einsatz der FabLab Anlagen im Rahmen der Themenstellung
- Präsentation der Workshop-Ergebnisse

Lernergebnisse:

Die Studierenden ...

- setzen eigene Fertigungsideen schnell, systematisch prototypischen um.
- kennen notwendige Schritte und Vorgehensweisen der Projektdurchführung und wenden diese auf einfache Projekte an.
- kennen weitere im FabLab verfügbare Anlagen und können sie angemessen und zielgerichtet einsetzen.
- beurteilen die Wirkungsweise und Begrenzungen ausgewählter Anlagen des FabLabs.
- können einschätzen, welche Werkstücke durch die im FabLab verfügbaren Anlagen in welcher Qualität herstellbar sind.
- können ihre Fähigkeiten für die Realisierung der Mechanik smarterer Geräte einsetzen.

Voraussetzungen und Empfehlungen:

Teilnehmende sollte in der Lage sein, technische Konstruktionen mit Hilfe von CAD-Programmen anfertigen zu können und insgesamt ein gutes räumliches und technisches Verständnis haben.

Literatur:

themenspezifisch

Studiengänge:
• Smart Technology Bachelor of Science

◆ MB223 – Machine Learning

Verantwortliche:	Hendrik Annuth
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB111 – Machine Learning
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	40 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	55 Stunden
Lehrende:	Hendrik Annuth

Teilleistung:	TB122 – Übg. Machine Learning
Lernform:	Vorlesung
Prüfungsform:	Abnahme
Prüfungsdauer:	120 Min.
ECTS:	3.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	75 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Hendrik Annuth

Studieninhalte:

- Einlesen, Bereinigen, Normalisieren und Erweitern von Datensätzen
- Durchführung von Regression, Classification und Auto Classifications Aufgaben
- Ergebnisoptimierung durch die Einstellung von Parametern innerhalb des Lernvorgangs
- Lerntypen: supervised, unsupervised, reinforcement, semi-supervised
- Aufgabentype: Regression, Classification, Auto Classification, Colaborative Filtering
- Überblick über die wichtigsten Metriken
- Datentypen und Verfahren: NLP, Computer Vision, Tabular Data
- Under- und Overfitting
- Gradient Decent
- Vanishing- und Exploding-Gradient-Problem
- Single Layered Perception, Multilayered Perception
- Learning Rate: Momentum, One Cycle Learning
- Optimierungen: SGD, Adam, RMSprop, Drop-Out, Batch Normalization, Weight Decay
- Convolutional Neural Networks
- Recurrent Neural Networks
- Common Machine Learning Libraries

Lernergebnisse:

- Welche Voraussetzungen haben Machine-Learning-Verfahren, wann, wie und wo ist deren Einsatz sinnvoll?
- Ein detailliertes Verständnis für die Vor- und Nachteile von Machine-Learning-Prozessen
- Das Gradientenabstiegsverfahren wird so detailliert behandelt, dass auch Änderungen und Erweiterungen des Verfahrens eingeordnet und verstanden werden können
- Der gekonnte Einsatz von Optimierungstechniken innerhalb des Lernprozesses
- Die Fähigkeit, Under- und Overfitting an deren jeweiligen Auswirkungen zu erkennen und sinnvolle Gegenmaßnahmen zu ergreifen
- Für verschiedene Problemstellungen, wie Bilderkennung, Textvervollständigung, Empfehlungssysteme und Klassifikationsprobleme eine sinnvolle künstliche neuronale Netzwerkarchitektur auswählen zu können
- Sicherer Umgang mit der Aufbereitung von Datensätzen für die Benutzung von Machine Learning
- Erfahrung mit der richtigen Auswahl einer künstlichen neuronalen Netzarchitektur für ein gegebenes Ziel mit gegebenen Daten

- Ein gutes Verständnis für die Auswahl und Einstellung von Optimierungsparametern zum Zwecke der Ergebnisoptimierung
- Erfahrung mit der Auswahl, Auswirkung und Bedeutung von verschiedenen Metriken

Voraussetzungen und Empfehlungen:

Das Modul Exploratory Data Analysis sind wichtige Voraussetzungen für das Modul. Erweitertes mathematisches Wissen aus dem Gebiet Statistik und Lineare Algebra wird in der Veranstaltung vorausgesetzt. Zusätzlich werden Fähigkeiten zur Programmierung in R und Python vorausgesetzt.

Literatur:

- The Elements of Statistical Learning; Springer 2009; Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython; O'Reilly and Associates 2012
- Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow; O'Reilly Media 2019; Aurélien Géron
- An Introduction to Statistical Learning; Springer 2013; Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani
- Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow; Packt Publishing 2019; Sebastian Raschka, Vahid Mirjalili
- Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play; O'Reilly UK Ltd. 2019; David Foster

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB345 – Architekturen vernetzter Systeme

Verantwortliche:	Hendrik Annuth
Moduldauer:	6 Monate
Unterrichtssprache:	None

Bestandteile:

Teilleistung:	TB297 – Workshop container- und serviceorientierte Architekturen
Lernform:	Workshop
Prüfungsform:	Teilnahme
Prüfungsdauer:	20 Min.
ECTS:	1.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	3 Termine
Zeit in Veranstaltungen:	7 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	10 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	12 Stunden
Gesamtaufwand:	29 Stunden
Lehrende:	Jan Sorgenfrei

Teilleistung:	TB339 – Workshop Cloud Infrastructure
Lernform:	Praktikum
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	35 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Hendrik Annuth Ulrich Hoffmann

Teilleistung:	TB340 – Computer-Netzwerke
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	6 Stunden
Aufwand während Semesterferien:	39 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Ilja Kaleck

Studieninhalte:

Das Modul vermittelt grundlegende und vertiefende Konzepte zur Architektur vernetzter, verteilter und cloubasierter Softwaresysteme. Der Fokus liegt auf service- und containerorientierten Architekturen, Cloud-Infrastrukturen sowie den zugrunde liegenden Computer-Netzwerken.

Im **Workshop Service- und containerorientierte Softwarearchitektur** werden DevOps-Prinzipien sowie der praktische Einsatz von Docker und Docker Compose zur Entwicklung, Bereitstellung und Wartung verteilter Anwendungen behandelt.

Der **Workshop Cloud Infrastructure** vertieft die Konzeption, Umsetzung und den Betrieb moderner Cloud-basierter Systeme. Behandelt werden Microservices-Architekturen, Containerisierung, DevOps-Ansätze, automatisiertes Testen, Test Driven Development, MLOps, AutoML sowie kontinuierliche Integrations- und Bereitstellungsprozesse. Die Inhalte werden projektbasiert erarbeitet und durch Zwischenpräsentationen begleitet.

Der Teilbereich **Computer-Netzwerke** vermittelt die technischen Grundlagen vernetzter Systeme, darunter Netzwerkarchitekturen, Kommunikationsprotokolle und grundlegende Mechanismen der Datenübertragung.

Studierende erhalten damit einen ganzheitlichen Überblick über Architektur, Betrieb und Vernetzung moderner Softwaresysteme und lernen, diese Konzepte praktisch anzuwenden und kritisch zu reflektieren.

Lernergebnisse:

Ziel des Moduls ist es, ein fundiertes Verständnis für Architekturen vernetzter Systeme zu entwickeln. Die Studierenden sind in der Lage, serviceorientierte und cloudbasierte Softwaresysteme zu entwerfen, umzusetzen und zu betreiben sowie deren technische und organisatorische Anforderungen zu bewerten.

Nach Abschluss des Moduls können die Teilnehmer containerbasierte Anwendungen entwickeln, Microservices-Architekturen einsetzen, DevOps- und MLOps-Konzepte anwenden und die zugrunde liegenden Netzwerkmechanismen nachvollziehen. Sie sind befähigt, komplexe vernetzte Systeme strukturiert zu analysieren und praxisorientierte Lösungen zu erarbeiten.

Voraussetzungen und Empfehlungen:

Empfohlen werden grundlegende Kenntnisse in der Softwareentwicklung sowie Basiswissen in Betriebssystemen und Programmierung. Erfahrungen mit verteilten Systemen oder Netzwerken sind von Vorteil, jedoch nicht zwingend erforderlich.

Literatur:

Martin Kleppmann: *Designing Data-Intensive Applications*, O'Reilly Media, 2017.
Sam Newman: *Building Microservices*, O'Reilly Media, 2015.
Andrew S. Tanenbaum, David J. Wetherall: *Computer Networks*, Pearson, 5th Edition, 2011.
Brendan Burns: *Designing Distributed Systems*, O'Reilly Media, 2018.

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB041 – Induktive Statistik

Verantwortliche:	Franziska Bönte
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB017 – Induktive Statistik
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Franziska Bönte

Studieninhalte:

- Wahrscheinlichkeitsrechnung
 - Grundlagen
 - Zufallsvariable und Wahrscheinlichkeitsverteilungen
 - Kombinatorik
 - Ausgewählte diskrete Verteilungen
 - Ausgewählte stetige Verteilungen
 - Hauptsätze der Wahrscheinlichkeitstheorie und Statistik
 - Approximationsregeln
- Stichproben
- Schätzverfahren
 - Punktschätzung
 - Intervallschätzung
- Testverfahren
 - Parametrische Testverfahren
 - Verteilungstests

Lernergebnisse:

Die Studierenden sind befähigt, weiterführende statistische Methoden zur Lösung komplexer Problemstellungen nutzen und die erzielten Ergebnisse korrekt zu interpretieren.

Die Studierenden erlangen ...

- Verteilungsannahmen für unterschiedliche ökonomische und naturwissenschaftliche Fragestellungen sinnvoll zu tätigen
- die Fähigkeit, Schätzwerte für die Parameter einer Grundgesamtheit zu bestimmen.
- die Fähigkeit, selbständig statistische Tests im Rahmen betrieblicher Aufgabenstellungen zu planen und durchzuführen und die Ergebnisse korrekt anzugeben.
- die Fähigkeit, Intervallwahrscheinlichkeiten unter Verwendung der wichtigsten diskreten und stetigen Dichte- und Verteilungsfunktionen zu berechnen.
- die Fähigkeit, Werte einer Grundgesamtheit zu schätzen und Hypothesen über die Werte einer Grundgesamtheit zu testen.
- die Fähigkeit, mittels geeigneter Computerprogramme statistische Untersuchungen großer Datenmengen vorzunehmen.

- Kenntnisse hinsichtlich des Einsatzes von Testverfahren im Rahmen der statistischen Qualitätskontrolle anhand von Problemstellungen aus der Wirtschaft.
- die Fähigkeit, sowohl eine Zeitreihe zu analysieren und die Komponenten einer Zeitreihe zu berechnen als auch kurz- und langfristige Prognosen durchzuführen.
- die Fähigkeit, die Genauigkeit von Prognosen kritisch zu bewerten.

Voraussetzungen und Empfehlungen:

keine Voraussetzungen, empfohlen werden aber Kenntnisse der deskriptiven Statistik.

Literatur:

- Bleymüller, Josef: Statistik für Wirtschaftswissenschaftler; 16. Aufl.; s.l.; Verlag Franz Vahlen; 2012
- Griffiths, William E.; Hill, R. Carter; Judge, George G.: Learning and practicing econometrics; ; New York; John Wiley; 1993
- Hansen, Gerd: Methodenlehre der Statistik; ; München; Vahlen; 1974
- Hansmann, Karl-Werner: Kurzlehrbuch Prognoseverfahren; ; Wiesbaden; s.l.; Gabler Verlag; 1983
- Lippe, Peter Michael von der: Wirtschaftsstatistik; 3., neubearb. u. erw. Aufl.; Stuttgart; Fischer; 1985
- Mood, Alexander MacFarlane; Boes, Duane C.; Graybill, Franklin A.: Introduction to the theory of statistics; 3. ed., international ed., [reprint.]; Auckland; McGraw-Hill; 2009
- Rüger, Bernhard: Induktive Statistik; 2., überarb. Aufl., 2. Nachdr; München; Oldenbourg; 1995
- Schlittgen, Rainer; Streitberg, Bernd H. J.: Zeitreihenanalyse; 3. Aufl., durchges. u. verb; München; R. Oldenbourg; 1989
- Zuckarelli, Joachim: Statistik mit R; ; Heidelberg; O'Reilly; 2017
- Bourier, Günther: Beschreibende Statistik. 11. Auflage. Wiesbaden: Springer Gabler, 2013.
- Kobelt, Helmut; Steinhausen, Detlef: Wirtschaftsstatistik für Studium und Praxis. 7. Auflage. Stuttgart: Schäfer-Poeschel Verlag, 2006.
- Schwarze, Jochen: Grundlagen der Statistik Band 2 : Wahrscheinlichkeitsrechnung und induktive Statistik. 10. Auflage. Berlin: nwb Studium 2013.
- Toutenburg, Helge u., a.: Induktive Statistik : Eine Einführung mit R und SPSS. 4. Auflage. Berlin: Springer-Verlag 2008.

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Smart Technology Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB087 – Systemmodellierung

Verantwortliche:	Dennis Proppe
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB032 – Systemanalyse
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	60 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	30 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Dennis Proppe

Teilleistung:	TB033 – Prozessmodellierung
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Christian Uhlig

Studieninhalte:

- Grundbegriffe der Systemanalyse
 - Gegenstand und Zielsetzung im Unternehmensumfeld
 - Methodische Grundlagen
 - Systemaufnahme
 - Rahmenbedingungen und Techniken der Informationsgewinnung
 - Thematische Untersuchungsbereiche
 - ER-Datenmodellierung
 - Systemmodellierung
 - Ereignisgesteuerte Prozessketten zur Modellierung von Geschäftsprozessen
 - Modellelemente schlanker EPK-Modelle und ihre Nutzung
 - Modellelemente erweiterter EPK-Modelle und ihre Nutzung
 - Business Process Model and Notation BPMN
 - Ausgewählte Modellelemente
 - Beispielmodelle
-
- Grundlagen des Geschäftsprozessmanagement
 - Motivation
 - Begriffe
 - Einordnung der Geschäftsprozessmodellierung
 - Bezüge zur Systemanalyse und zum Software-Engineering
 - ARIS-Methode
 - Sichtenkonzept
 - Schichtenkonzept
 - Überblick über Modelltypen und ihre Vernetzung
 - ARIS-Softwaresystem
 - Modellierung der Aufbauorganisation
 - ER-Datenmodellierung
 - Funktionsmodellierung
 - Prozessmodellierung
 - Wertschöpfungsketten (WSK) und Prozesslandkarten
 - EPK
 - Vernetzung mit anderen ARIS-Sichten (Daten, Aufbauorganisation)
 - BPMN (Process und Collaboration Diagrams)
 - Praktische Aufgabenstellungen

- Ausschnittsweise und formfreie Modellierung von Prozessen aus einem beispielhaften Fachkonzept
- Modellierung des Datenmodells zu einem beispielhaften Fachkonzept (ERD)
- Ausschnittsweise Modellierung von Prozessen zu einem beispielhaften Fachkonzept (WSK / EPK und BPMN)
- Ganzheitliche Modellierung von Aufbauorganisation, Daten und Prozessen zu einer Fallstudie (Organigramm, ERD, WSK, BPMN)

Lernergebnisse:

Die Studierenden ...

- beurteilen die generellen Möglichkeiten und Grenzen von Systemanalysen, insbesondere in Bezug auf die Gestaltung betrieblicher Informationssysteme.
- unterscheiden die wesentlichen Techniken zur Informationsgewinnung in Unternehmen einschließlich ihrer Vor- und Nachteile, bewerten Techniken im Kontext einer konkreten Informationsgewinnung.
- führen eine methodisch fundierte Informationsgewinnung in einem überschaubaren Problemkontext durch.
- erklären wichtige Bestandteile und Schritte der Systemaufnahme als Vorphase zur Systemmodellierung, sie setzen ausgewählte Formalismen zur Dokumentation der Aufnahmeergebnisse ein.
- stellen die im Unternehmensumfeld praktisch relevanten methodischen Ansätze zur Systemmodellierung dar und beurteilen diese hinsichtlich ihrer Eignung für bestimmte Erkenntnisziele im Rahmen einer Systemanalyse.
- beschreiben die zu den methodischen Ansätzen gehörenden Modellnotationen und setzen diese angemessen zur Modellierung ein.
- nutzen die Modellierungsmittel zum Aufbau von Analysemodellen begrenzter Komplexität für betriebswirtschaftlich ausgerichtete Informationssysteme und diesbezügliche Problemstellungen.

Die Studierenden ...

- nennen und erläutern die theoretischen Grundlagen des Geschäftsprozessmanagement und seine Begriffswelt.
- nennen und erläutern die Grundlagen der Geschäftsprozessmodellierung sowie ihre Einordnung in das Geschäftsprozessmanagement und andere Themenbereiche wie Softwareengineering, Datenbanken und Systemanalyse.
- stellen die Grundlagen der ARIS-Methode dar.
- wenden wesentliche Modellierungskonzepte des ARIS-Softwaresystems (insbesondere Objekte und Kanten, Definitions- und Ausprägungsebenen sowie Hinterlegungen) an.
- nutzen das ARIS-Softwaresystem in seinen wesentlichen Bedienkonzepten und -elementen zur Erstellung von miteinander vernetzten Modellen.
- analysieren komplexe textuelle Fachkonzeptbeschreibungen und unterscheiden dabei Inhalte der verschiedenen ARIS-Modellierungssichten (Organisation, Daten, Leistungen, Funktionen, Steuerung).
- entwerfen und gestalten in ästhetisch ansprechender Weise Modelle zentraler Modelltypen (ER-Modell, EPK, WSK, BPMN Process und Collaboration, Organigramm) zu komplexen Fachkonzeptbeschreibungen.

Voraussetzungen und Empfehlungen:

- Grundverständnis der Informationstechnologie
- Kenntnis in objektorientierten Programmiersprache(n)
- Grundsätzliches Verständnis betriebswirtschaftlicher Zusammenhänge

Literatur:

- KRALLMANN, H.; BOBRIK, A.; LEVINA, O.: Systemanalyse im Unternehmen - Prozessorientierte Methoden der Wirtschaftsinformatik, Oldenbourg, 2013
- ALLWEYER, Thomas: BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung. 2. Auflage, Books on Demand, 2020
- GADATSCH, Andreas: Grundkurs Geschäftsprozess-Management: Analyse, Modellierung, Optimierung und Controlling von Prozessen. 9. Aufl., Springer Vieweg, 2020
- RUPP, Chr.: Systemanalyse kompakt, Springer Verlag, 2013
- HÄUSLEIN, A.: Systemanalyse. vde-Verlag, 2004
- KRÜGER, J.; UHLIG, Ch.: Praxis der Geschäftsprozessmodellierung. VDE Verlag, 2009
- SCHEER, A.-W.: Architektur integrierter Informationssysteme. Springer-Verlag, Berlin, 1991
- OBJECT MANAGEMENT GROUP OMG: Business Process Model and Notation (BPMN), Version 2.0, URL: <http://www.omg.org/spec/BPMN/2.0>, 2011
- FREUND, Jakob ; RÜCKER, Bernd: Praxishandbuch BPMN 2.0. 4. aktualisierte Auflage, Carl Hanser Verlag, 2014
- OESTERREICH, B.: Analyse und Design mit UML 2.3: Objektorientierte Softwareentwicklung. Oldenbourg, 2009
- KECHER, Chr.: UML 2: Das umfassende Handbuch. Galileo Computing, 2011

-
- Krüger, Jörg; Uhlig, Christian: Praxis der Geschäftsprozessmodellierung - ARIS erfolgreich anwenden, VDE Verlag, 2009
 - Lehmann, Frank: Integrierte Prozessmodellierung mit ARIS, dpunkt.verlag, 2007
 - Scheer, August-Wilhelm: ARIS Vom Geschäftsprozess zum Anwendungssystem, 4. Auflage, Springer, 2002
 - Scheer, August-Wilhelm: ARIS Modellierungsmethoden, Metamodelle, Anwendungen, 4. Auflage, Springer, 2001
 - OMG: Business Process Model and Notation (BPMN) Version 2.0, 2011
 - Software AG: ARIS-Dokumentation (Methodenhandbuch, Bedienhandbücher), jeweils aktuellste Fassung

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Smart Technology Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB116 – Technologie der Mediengestaltung und GUI-Programmierung

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB089 – Technologie der Mediengestaltung und GUI-Programmierung
Lernform:	Vorlesung
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	60 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	90 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Ole Nass

Studieninhalte:

- Motivation, Begriffe und Konzepte
 - Mensch-Computer-Interaktion (MCI): Ziele, Herausforderungen, Modelle
 - Überblick, Technologien webbasierter Programmierung
- Menschliche Informationsverarbeitung und ihre Bedeutung für die MCI
 - Modelle zur Informationsverarbeitung
 - Sinne und ihre Relevanz
 - Wahrnehmungsgesetze und Gedächtnis
 - Handlungspsychologie und das Interface als Handlungsraum
 - Handlungsprozesse und Fehlerbehandlung
- Interaktion im Dialog
 - Funktions- und ablauforientierte Interaktion
 - Gestaltungsgrundsätze
 - Wahrnehmungsbasierte Organisation komplexer Informationen
 - Navigation in multimedialen Anwendungen
 - Normen, Gesetze, Richtlinien
 - Barrierefreiheit
- Technologien für Dynamische Webseiten
 - Client-Server-Modell
 - Frameworks
 - Begriffe: Mandantenfähigkeit, Backend, Backoffice, Frontend, Template
- Konkrete Technologien für unterschiedliche Clients
 - Auswahl aktueller Technologien
 - Übungen: z. B. HTML5, CSS, AJAX, JavaScript, Webapp, Flash, Flex / Air...

Lernergebnisse:

Studierende erhalten ...

- die Fähigkeit, aktuelle Technologien der Programmierung von webbasierten Medien funktional und operativ zu durchdringen,
- die Fähigkeit, die Aspekte, mit denen spezielle Gestaltungsvorstellungen umzusetzen sind, zu erkennen,

- das Verständnis der physiologischen und psychologischen Grundkonzepte von Interaktionen, das die Zusammenhänge zwischen menschlicher Informationsverarbeitung und Konzepten zur Analyse und Gestaltung interaktiver Systeme transparent macht,
- das Verständnis der softwareergonomischen Richtlinien / Normen zu den Informationstechnik-Verordnungen zur Barrierefreiheit sowie
- das Verständnis der unterschiedlichen Hardwarekonzepte für interaktive Ein- und Ausgabemedien.

Voraussetzungen und Empfehlungen:

- Vertrautheit mit webbasierten Technologien wie HTML5, CSS und JavaScript.
- Fähigkeit zur Entwicklung dynamischer Webseiten und Anwendungen.

Literatur:

- Böhringer, J., Bühler, P., Schlaich, P., Sinner, D. (2014): Kompendium der Mediengestaltung, 6. Auflage, Berlin: Springer Verlag (4 Bände: Konzeption, Technik, Print, Digital)
- Butz, A., Krüger, A. (2014): Mensch-Maschine-Interaktion. München: De Gruyter/ Oldenbourg
- Dahm, M. (2006): Grundlagen der Mensch-Computer-Interaktion. München: Pearson Studium
- Hammer, N., Bensmann, K. (2011): Webdesign für Studium und Beruf. Webseiten planen, gestalten und umsetzen. 2. Aufl. Berlin: Springer Verlag.
- Heinecke, A.M. (2012): Mensch-Computer-Interaktion. 2. Aufl. Berlin: Springer
- Khazaeli, C.D. (2005): Systemisches Design, Intelligente Oberflächen für Information und Interaktion. Reinbek: Rowohlt Verlag
- Malaka, R., Butz, A., Hußmann, H. (2009): Medieninformatik. Eine Einführung. München: Pearson Studium
- Preim, B., Dachzelt, R. (2015): Interaktive Systeme, Band 1+2. Berlin: Springer Vieweg
- Sharp, Helen; Rogers, Yvonne; Preece, Jenny (2007): Interaction Design: Beyond Human-Computer Interaction. 2nd edition, New York: Wiley.
- Stephanidis, Constantine (ed) (2009): Universal Access in Human Computer Interaction. Intelligent and Ubiquitous Interaction Environments. 5th International Conference UAHCI 2009, San Diego, CA, USA (LNCS 5615). Berlin, New York: Springer

Studiengänge:

- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB135 – Projekt Eingebettete Systeme

Verantwortliche:	Timm Bostelmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB196 – Prakt. PCB-Design
Lernform:	Praktikum
Prüfungsform:	Abnahme
Prüfungsdauer:	20 Min.
ECTS:	1.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	3 Termine
Zeit in Veranstaltungen:	11 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	19 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	30 Stunden
Lehrende:	Timm Bostelmann

Teilleistung:	TB198 – Prakt. Schaltungstechnik
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	
ECTS:	1.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	4 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	30 Stunden
Lehrende:	Thomas Starke

Teilleistung:	TB199 – Projekt Mikrocontroller
Lernform:	Projektarbeit
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	30 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Timm Bostelmann

Studieninhalte:

Projekt Mikrocontroller:

- Praktische Unterweisung an den Laborgeräten
- Planung der Projektdetails und des Projektablaufs
- Hardware-Schaltungsentwurf inklusive eines Mikrocontrollers
- Softwareerstellung für den eingesetzten Mikrocontroller
- Inbetriebnahme der Schaltung auf einem Steckbrett
- Test und empirischer Nachweis der Funktion
- Technische Dokumentation des Projektes

Prakt. PCB-Design:

- Leiterplattentechniken
- Einführung in ein PCB-Tool
- Durchführung eines Leiterplattenentwurfs
- Fertigung der Leiterplatte
- Inbetriebnahme der Leiterplatte

Prakt. Schaltungstechnik:

- Wechselnde, ausgewählte Versuche aus dem Stoffgebiet der Vorlesungen Elektronik und Halbleiterschaltungstechnik

Lernergebnisse:

Die Studierenden...

- können sich in ein vorgegebenes oder selbst gewähltes Projektthema einarbeiten und dessen Realisierung planen.
- können analoge und digitale Hardware (je nach Projektthema) als Schnittstelle eines Mikrocontrollers entwerfen.
- können Software für einen Mikrocontroller entwickeln, testen und in Betrieb nehmen.
- können ihre Lösung auf System- und Implementierungsebene in einer schriftlichen Ausarbeitung dokumentieren.
- kennen Entwurfsregeln für Leiterplatten.
- sind in der Lage sich in eine Leiterplattenentwurfssoftware einzuarbeiten.
- können eine Leiterplatte entwickeln und bestücken.
- kennen Regeln zum Umgang mit elektronischen Komponenten.
- kennen Vor- und Nachteile unterschiedlicher Messmethoden.
- können Messreihen durchführen, protokollieren und auswerten.

Voraussetzungen und Empfehlungen:

Kenntnisse in Elektronik, Halbleiterschaltungstechnik und systemnaher Programmierung mit C werden vorausgesetzt.

Literatur:

Datenblätter der verwendeten Komponenten (Projektabhängig)
Graham Keith, Jon Evans, Glenn Peterson: Getting Started in KiCad, The KiCad Team, 2023
Ulrich Tietze, Christoph Schenk: Halbleiterschaltungstechnik, 16. Auflage Springer Verlag, 2016

Studiengänge:

- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB244 – Exploratory Data Analysis

Verantwortliche:	Hendrik Annuth
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB079 – Data Visualization, Feature Engineering
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	75 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Hendrik Annuth

Teilleistung:	TB090 – Übg. Exploratory Data Analysis
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Marco Pawlowski

Studieninhalte:

- Machine-Learning-Verfahren Naive Bayes und Random Forest
- Data Transformation
- Data Augmentation
- Feature Embedding
- Feature Analysis: Feature Importance, Partial Dependencies; Data Leakage
- Unstructured Data
- Zeitreihen
- Erweiterte Einführung in R und Python
- Wichtigste Pakete zur Datenverarbeitung in R und Python
- Verwendung der gängigsten Visualisierungsverfahren in R und Python
- Erstellung von Diagrammen, Graphen und Histogrammen
- Zusammenstellung von Informationsgrafiken zur gleichzeitigen Darstellung
 - Das Pyramidenprinzip
 - Auswahlkriterien für Balken-, Torten- und Streudiagramme, Histogramme, Graphen
 - Metriken und Dashboards
 - Einführung in die Datenreinigung, -aufbereitung und -erweiterung
 - Einführung in die Topologie
 - Automatisierte Visualisierungsanwendungen
 - Clustering Methoden k-Means, Self-organizing Maps, Hierarchical Clustering
 - Dimensionsalitätsreduktionsverfahren: PCA, T-SNE, UMAP

Lernergebnisse:

- Ein gutes Verständnis, wie beliebige Daten in einen Vektor umgewandelt bzw. als solches aufgefasst werden können, um verschiedene Datenarten für den Einsatz von Machine-Learning-Verfahren nutzbar zu machen
- Sicheres Beherrschen von Verfahren zur Datenreinigung, Data Augmentation und Datentransformation, um bewusst Problemen im Lernprozess vorzubeugen

- Kenntnis von Normalisierungstechniken, um die Ergebnisse von Machine-Learning-Verfahren zu verbessern
- Ein solides Verständnis von kategorischen und kontinuierlichen Datentypen, deren Konvertierung, von Einstellungsparametern und deren Auswirkungen auf ein Machine-Learning-Verfahren
- Ein Verständnis und Lösungsansätze beim Umgang mit Daten, die Zeitreihen enthalten
- Einsatz von Visualisierungstechniken, um aus Datensätzen verbesserten inhaltlichen Nutzen ziehen zu können
- Bewusstsein für die gezielte Auswahl von Graphen, Diagrammen und Tabellen zum Zwecke der Hervorhebung von Erkenntnissen aus Datensätzen
- Grundverständnis für die Funktionsweise aktueller Softwarelösungen und Pakete für die Visualisierung und Aufbereitung von Daten
- Erschließung von Datensätzen und deren Besonderheiten durch die Programmiersprachen R und Python
- Die Fähigkeit, Visualisierungstechniken im Kontext der Datensatzanalyse so einzusetzen, dass Datensätze durch den Erkenntnisgewinn sinnvoll aufbereitet und erweitert werden können
- Verständnis zur Visualisierung hochdimensionaler Daten

- Sicherer Umgang mit der Programmierung in R und Python und den dazugehörigen Entwicklungsumgebungen

- Erfahrung mit dem Laden, Verarbeiten und Visualisieren von komplexen Datensätzen

- Fähigkeit zur programmatischen Erstellung von verschiedenen Diagrammen, Graphen und Histogrammen

- Erfahrung mit dem Zusammenstellen verschiedener Informationsgrafiken im Kontext der Aufbereitung einer zielgerichteten Fragestellung

Voraussetzungen und Empfehlungen:

Das Modul setzt erweiterte Programmierkenntnisse voraus. Sprachelemente aus Python und R werden vorgestellt und verwendet. Dabei werden Konzepte wie Zeiger, Speicherreservierung und Garbage-Collection als bekannt vorausgesetzt. Mathematisches Grundwissen aus den Veranstaltungen Mathematik 1 und 2, aus den Gebieten Statistik und Linearer Algebra wird in der Veranstaltung vorausgesetzt.

Literatur:

- Feature Engineering and Selection: A Practical Approach for Predictive Models; Chapman and Hall 2019; Max Kuhn and Kjell Johnson;
- Feature Engineering for Machine Learning Models: Principles and Techniques for Data Scientists; O'Reilly 2018; Alice Zheng, Amanda Casari
- The Art of Feature Engineering Essentials for Machine Learning; Cambridge University Press 2020; Pablo Duboue

Siehe auch Veranstaltung Machine Learning

- Visual Display of Quantitative Information; Bertrams 2001; Edward R Tufte
- Say It With Charts: The Executives's Guide to Visual Communication: The Executive's Guide to Visual Communication; McGraw-Hill ; Gene Zelazny
- The Pyramid Principle: Logic in Writing and Thinking: Logical Writing, Thinking and Problem Solving; Financial Times Series 1996; Barbara Minto
- The Elements of Statistical Learning; Springer 2009; Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Datenvisualisierung: Vom Diagramm zur Virtual Reality; UTB 2018; Peter Fischer-Stabel

- Storytelling mit Daten: Die Grundlagen der effektiven Kommunikation und Visualisierung mit Daten; Vahlen 2017; Cole Nussbaumer Knaflitz, Mike Kauschke
- The Truthful Art: Data, Charts, and Maps for Communication (Voices That Matter); New Riders 2016; Alberto Cairo
- Infografik: Komplexe Daten professionell visualisieren; Rheinwerk Design 2018; Raimar Heber
- Data Visualization: A Practical Introduction; Princeton University Press 2019; Kieran Healy
- Datenvisualisierung mit Tableau; mitp 2018; Alexander Loth
- Learning Python; O'Reilly and Associates 2013; Mark Lutz
- Basic Elements of Computational Statistics; Springer 2017; Wolfgang Karl Härdle, Ostap Okhrin, Yarema Okhrin
- Data Science mit Python: Das Handbuch für den Einsatz von IPython, Jupyter, NumPy, Pandas, Matplotlib und Scikit-Learn; mitp 2017; Jake VanderPlas

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB293 – Digital Product Management

Verantwortliche:	Atilla Wohllebe
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB284 – Digital Product Management
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	60 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Atilla Wohllebe

Studieninhalte:

- Einführung Digital Product Management
- Konzeption von Produkten
 - Erarbeitung einer Product Vision
 - Lösungsentwicklung im Rahmen der Product Discovery (Design Sprint)
 - Validierung von Produktideen am Markt
- Datengetriebene Produktentwicklung
 - Entwicklung und Weiterentwicklung von Produkten
 - Agile Arbeitsweisen (Scrum, Kanban)
 - Erhebung und Priorisierung von Anforderungen
 - DevOps - Grundlegende Gedanken und Methoden

Lernergebnisse:

Die Studierenden...

- verfügen über weitgehende Kenntnisse zum Management digitaler Produkte und Services, insbesondere mit Blick auf deren konzeptionelle (Weiter-) Entwicklung.
- sind in der Lage, mit Hilfe geeigneter Methoden eine Product Vision zu entwickeln und diese im Rahmen einer Product Discovery in konkrete Lösungsansätze zu übersetzen und zu validieren.
- verfügen außerdem über Kenntnisse agiler Arbeitsweisen und aktueller Vorgehensweisen in der Softwareentwicklung und können diese in den Kontext des Digital Product Managements einordnen.

Voraussetzungen und Empfehlungen:

Keine

Literatur:

- - Düsterbeck, F., Einemann, I. (2022). Product Ownership meistern - Produkte erfolgreich entwickeln. dpunkt.verlag, Heidelberg.

- Hoffmann, S. (2020). Digitales Produktmanagement: Methoden - Instrumente - Praxisbeispiele. Springer Gabler, Wiesbaden. <https://link.springer.com/book/10.1007/978-3-658-30629-8>
- Kim, G., Humble, J., Debois, P., Willis, J., Forsgren, N. (2022). Das DevOps Handbuch - Teams, Tools und Infrastrukturen erfolgreich umgestalten. O'Reilly / dpunkt.verlag, Heidelberg.
- Kittlaus, H., Clough, P. (2009). Software Product Management and Pricing - Key Success Factors for Software Organizations. Springer Berlin, Heidelberg. <https://link.springer.com/book/10.1007/978-3-540-76987-3>
- Pranam, A. (2018). Product Management Essentials - Tools and Techniques for Becoming an Effective Technical Product Manager. Apress Berkeley, CA. <https://link.springer.com/book/10.1007/978-1-4842-3303-0>
- Wagenblatt, A. (2019). Software Product Management - Finding the Right Balance for YourProduct Inc.. Springer Cham. <https://link.springer.com/book/10.1007/978-3-030-19871-8>

Studiengänge:

- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB356 – Projekt Connected Devices

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB350 – Projekt Connected Devices
Lernform:	Workshop
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	30 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	30 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	75 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

Grundlagen der Kommunikation zwischen vernetzten smarten Geräten (Connected Devices); Drahtlose Übertragungstechnologien (WLAN/Wi-Fi, LoRaWAN, Zigbee) und deren Charakteristika hinsichtlich Reichweite, Energieverbrauch und Datenrate; Kommunikationsprotokolle für IoT-Systeme (MQTT, CoAP, HTTP/REST, RFC 8949 CBOR); Architektur von IoT-Systemen mit Web-Servern, Gateways und Cloud-Anbindung; Praktische Realisierung eines vernetzten Systems im Rahmen einer Semesteraufgabe.

Lernergebnisse:

Die Studierenden ...

- kennen die wesentlichen drahtlosen Kommunikationstechnologien für vernetzte Geräte (WLAN, LoRaWAN, Zigbee) und können deren Eigenschaften hinsichtlich Reichweite, Energieverbrauch, Datenrate und Einsatzgebiet vergleichen und bewerten.
- verstehen den Aufbau und die Funktionsweise schichtbasierter IoT-Kommunikationsarchitekturen und können geeignete Protokolle (MQTT, CoAP, HTTP, CBOR) für konkrete Anwendungsfälle auswählen und begründen.
- sind in der Lage, ein System mit mehreren vernetzten Geräten einschließlich Web-Server, Gateway und Datenaustausch eigenständig zu konzipieren und zu realisieren.
- haben praktische Erfahrung in der Inbetriebnahme und dem Testen von IoT-Kommunikationsstrecken mit realer Hardware gesammelt.
- können ihre erworbenen Kenntnisse und Fähigkeiten kompetent in eine Gruppenleistung einbringen und dabei inhaltliche sowie organisatorische Schnittstellen im Team koordinieren.
- besitzen vertiefte Kenntnisse aktueller Standards und Spezifikationen im Bereich Connected Devices und können diese kritisch einordnen.

Voraussetzungen und Empfehlungen:

Für eine erfolgreiche Teilnahme werden Grundkenntnisse in der Programmierung eingebetteter Systeme sowie Grundlagenwissen in Rechnernetzen (TCP/IP, HTTP) vorausgesetzt. Erfahrungen mit Mikrocontroller-Plattformen (z. B. Arduino, ESP32, Raspberry Pi) und erste Kenntnisse in der Softwareentwicklung für smarte Systeme sind empfehlenswert.

Literatur:

- Hanes, D., Salgueiro, G., Grossetete, P., Barton, R., Henry, J.: *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press, Indianapolis, 2017. ISBN 978-1-58714-456-1.
- Wiegmann, B.: *IoT Networks with LoRaWAN*. Leanpub, 2022.
- Kamath, A.: *Beginning LoRa Radio Networks with Arduino*. Apress/Springer Nature, 2019. ISBN 978-1-4842-4356-5.

Studiengänge:

- Smart Technology Bachelor of Science

◆ MB366 – Datenbanken

Verantwortliche:	Marco Pawlowski
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB361 – Datenbanken
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	60 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	30 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 45 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Marco Pawlowski

Teilleistung:	TB362 – Übg. Datenbanken
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	20 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	1 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	7 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	52 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	59 Stunden
Lehrende:	Lars Neumann

Studieninhalte:

Vorlesungsbegleitende praktische Übungen in SQL und zum Datenbankentwurf

- Einführung in die Datenbanktechnologie
- Datenbanksprache SQL - Einführung
- Datenbank-Abfrage mit SQL
- Datenbanksprache SQL - Einrichten der Datenbank
- Das Entity-Relationship-Datenmodell
- Das Relationale Datenmodell
 - Relationenschemata und Datenabhängigkeiten
 - Relationale Datenbanken
 - Normalformen
- Datenbank - Lebenszyklus

Lernergebnisse:

Die Studierenden ...

- besitzen die Fähigkeit, ein Datenbanksystem mit SQL zu befragen und in nicht-triviale textuelle Anfrageanforderungen in SQL zu überführen.
- haben grundlegende Kenntnisse über die Ausführung der von ihnen gestellten Anfragen.
- haben die Kompetenz, ein Datenbankentwurfswerkzeug grundlegend zu bedienen.

Die Studierenden ...

- beherrschen die Grundlagen der relationalen Datenbanktechnologie;
- erlangen die Fähigkeit, selbstständig einen Datenbankentwurfsprozess zu planen, eine relationale Datenbank unter Nutzung von SQL einzurichten und die Informationsverarbeitung mittels relationaler Datenbanksysteme unter Nutzung von SQL durchzuführen;
- erlangen die Fähigkeit, mit einem Entwurfstool einen Datenbankentwurfsprozess durchzuführen und mittels SQL selbstständig Anfragen an ein Datenbanksystem zu stellen.

Voraussetzungen und Empfehlungen:

Vorausgesetzt wird ein grundlegendes Verständnis der Konzepte von Programmiersprachen.

Empfohlen wird die Einrichtung der in der Übung verwendeten Werkzeuge.

Literatur:

- Elmasri, Ramez; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen. 3. Aufl. München: Pearson -Verlag, 2009.
- Meier, Andreas: Relationale Datenbanken Leitfaden für die Praxis. Berlin: Springer-Verlag, 2004.
- Vetter, Max: Aufbau betrieblicher Informationssysteme mittels konzeptioneller Datenmodellierung. 8. Aufl. Stuttgart: Vieweg-Teubner, 1998.
- Vossen, Gottfried: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. 5. Aufl. Oldenbourg: Oldenbourg-Wissenschaftsverlag, 2008.

Vorlesungsunterlagen

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB373 – Robotik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB086 – Prakt. Robotik
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	10 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	15 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 105 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Hermann Höhne

Teilleistung:	TB368 – Robotik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	15 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

Inhalte der Vorlesung:

- Strukturen der Fertigungstechnik
- Flexible Fertigungszellen
- Industrieroboter
- Strukturen und Aufbau von Robotern
- Kinematik
- Antriebe
- Effektoren
- Steuerstrategien
- Koordinatentransformationen
- Punkt-zu-Punkt-Steuerung
- Steuerung mit Interpolation
- Mensch-Maschine-Kommunikation
- Roboter-Programmiersysteme
- Roboter-Sprachen
- Intelligente Sensorik
- Integration von Optischen Sensoren

Anhand eines Projekts werden die Inhalte aus der Vorlesung praktisch umgesetzt. Schwerpunkt liegt bei den Grundlagen zur Programmierung mobiler Roboter bis hin zu einfachem autonomen Fahren.

Lernergebnisse:

Die Studierenden ...

- besitzen fundierte Kenntnisse der technischen Grundlagen der Robotik.
- besitzen ausgehend von den Entwicklungstendenzen im Bereich der Flexiblen Fertigungstechnik die grundlegende Kompetenz für das Verständnis für Funktionsweisen und Einsatzschwerpunkte von Industrierobotern.
- verfügen über die Kompetenz, die Möglichkeiten der Verbindung von Robotern mit "intelligenten" Sensoren zu durchdringen, insbesondere die Erkennung und Einschätzung der Eigenschaften optischer Sensorsysteme.
- können verschiedene Konzepte der Offline-Programmierung von Industrierobotern identifizieren.

- verstehen aktuelle Entwicklungstendenzen zur Erhöhung der Selbständigkeit bei Robotern.

Nach Bearbeitung des Praktikums sind die Studierenden in der Lage ...

- Lerninhalte der Vorlesung im Rahmen eigener Erfahrungen zu vertiefen.
- Sich selbstständig in eine neue Programmierungsumgebung einzuarbeiten.
- Ein gegebenes Roboterprogrammiersystem zu nutzen.
- Techniken zum Sammeln, Glätten und Bewerten von Sensordaten anzuwenden.
- Typische Problemstellungen mittels einfacher Algorithmen zu lösen.
- Eigene Ideen und Lösungsansätze zu implementieren.
- Versuchsergebnisse in einer schriftlichen Dokumentation zu präsentieren.

Voraussetzungen und Empfehlungen:

- Kenntnisse in Linearer Algebra und Trigonometrie
- Grundlegende Fähigkeiten imperative Programme zu erstellen und auf Software-Bibliotheken zuzugreifen
- Grundkenntnisse und grundlegende Fähigkeiten in der Programmierung von Bildverarbeitungsalgorithmen und der Benutzung einschlägiger Bibliotheken.

Literatur:

- McKerrow: Introduction to Robotics, Addison Wesley, 1990
- Wirth: Flexible Fertigungssysteme, Hüthig-Verlag
- Vukobratovic: Introduction to Robotics, Springer, 1995
- Blume, Dillmann: Frei Programmierbare Roboter, Vogel Verlag
- Blume, Jakob: Programmiersprachen für Industrieroboter, Vogel Verlag, 1985
- Hertzberg, J: "Mobile Roboter: Eine Einführung aus Sicht der Informatik", eXamen.press, 2012
- Prat: "Sensordatenfusion und Bildverarbeitung zur Objekt- und Gefahrenerkennung", 2010
- Quigley: "Programming Robots with ROS", 2016

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB376 – Projekt Intelligente Systeme

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	None

Bestandteile:

Teilleistung:	TB372 – Projekt Intelligente Systeme
Lernform:	Projektarbeit
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	15 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	50 Stunden
Aufwand während Semesterferien:	50 Stunden
Flexibel einteilbarer Aufwand:	170 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 300 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.</i>
Lehrende:	Ulrich Hoffmann

Studieninhalte:

themenabhängig

Lernergebnisse:

Die Studierenden ...

- haben die Fähigkeit einfache Intelligente Systeme in Hard- und Software zu konzipieren, entwerfen und zu realisieren.
- kennen den Beziehung zwischen den intelligenten (smarten) Anteilen eines Systems und den für ihre Realisierung notwendigen low-level Komponenten.
- besitzen weitere Erfahrungen mit betreuter Projektarbeit im kreativen Umfeld
- verfügen über eine vertieftes theoretisches Wissen über intelligente Systeme, das durch praktischen Projektarbeit gefestigt ist.
- können ihre erworbenen Kenntnisse und Fähigkeiten kompetent in eine Gruppenleistung einbringen.
- haben vertiefte Kenntnisse der Bedeutung von inhaltlichen und organisatorischen Schnittstellen im Rahmen der Arbeit an einer Gesamtthematik.

Voraussetzungen und Empfehlungen:

Für eine erfolgreiche Teilnahme werden fundierte Kenntnisse in der Konzeption und Realisierung eingebetteter Systeme sowie Erfahrungen in der Projektarbeit und Grundkenntnisse in Hard- und Softwareentwicklung smarter Systeme vorausgesetzt.

Literatur:

themenabhängig

Studiengänge:

- Smart Technology Bachelor of Science
-

◆ MB378 – Fortgeschrittene Lineare Algebra

Verantwortliche:	Andreas Haase
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB374 – Fortgeschrittene Lineare Algebra
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Andreas Haase

Studieninhalte:

- Determinanten
 - Definition, Beispiele und Eigenschaften
 - Der Entwicklungssatz von Laplace
 - Cramer'sche Regel und lineare Gleichungssysteme
- Vektorräume
 - Definition, Beispiele und Eigenschaften
 - Unterräume
 - Lineare (Un-)Abhängigkeit, Basis und Dimension
- Euklidische und unitäre Vektorräume
 - Skalarprodukt und Norm
 - Orthogonalität
 - Orthogonal- und Orthonormalbasen
- Analytische Geometrie
 - Darstellung von Geraden und Ebenen
 - Lagebeziehung zwischen linearen geometrischen Objekten
 - Einfache nichtlineare Objekte an Beispielen
- Abbildungen
 - Lineare Abbildungen
 - Affine Abbildungen
 - Koordinatentransformationen
- Eigenwerte und Eigenvektoren
 - Charakteristisches Polynom, Eigenwerte, Eigenvektoren
 - Diagonalisierung von Matrizen
 - Matrixfunktionen

Lernergebnisse:

Nach dem erfolgreichen Besuch der Veranstaltung können die Studierenden ...

- die Determinante einer Matrix beliebiger Dimension berechnen und den Zusammenhang zur Lösungstheorie linearer Gleichungssysteme herstellen.
- die Vektorraumaxiome nennen und eine gegebene Menge mit Verknüpfungen darauf überprüfen, ob diese ein Vektorraum (über $\mathbb{R}$ oder $\mathbb{C}$) ist.

- Die Definition eines Unterraums nennen; Teilmengen von Vektorräumen darauf überprüfen, ob diese Unterräume sind.
- das Konzept der linearen Abhängigkeit von Vektoren erklären; Teilmengen von Vektorräumen auf lineare Abhängigkeit überprüfen.
- die Definition einer Basis nennen. Teilmengen von Vektorräumen darauf überprüfen, ob diese eine Basis sind.
- die Definition eines Skalarproduktes nennen. Verschiedene lineare Abbildungen auf Vektorräume darauf überprüfen, ob diese ein Skalarprodukt sind.
- die Definition einer Norm nennen; den Zusammenhang zwischen Skalarprodukt und Norm nennen.
- die Definition einer Orthonormalbasis nennen; eine Orthonormalbasis aus einer gegebenen Basis konstruieren (Gram-Schmidt-Verfahren).
- die Parameter und Koordinatendarstellung von Geraden und Ebenen formulieren; Lagebeziehungen zwischen linearen geometrischen Objekten berechnen; Lagebeziehungen zwischen linearen und einfachen nichtlinearen Geometrischen Objekten berechnen.
- die Definition einer linearen Abbildung nennen; lineare Abbildungen mittels Matrix-Vektor-Schreibweise ausdrücken. Eigenschaften gegebener linearer Abbildungen bestimmen.
- die Definition einer affinen Abbildung nennen; affine Abbildungen mittels Matrix-Vektor-Schreibweise ausdrücken. Eigenschaften gegebener affiner Abbildungen bestimmen.
- Koordinatentransformationen als affine Abbildung durchführen; die affine Abbildung einer Koordinatentransformation berechnen; aktive und passive Koordinatentransformationen unterscheiden.
- das charakteristische Polynom einer Matrix aufstellen; die Eigenwerte einer Matrix berechnen; die Eigenvektoren einer Matrix berechnen.
- eine Matrix diagonalisieren.
- bestimmte Funktionen einer Matrix berechnen.

Voraussetzungen und Empfehlungen:

Es werden grundlegende Kenntnisse zu Vektoren, Vektoralgebra, Matrizen, Matrixalgebra, Lösen von linearen Gleichungssystemen mittels Gauß-Verfahren vorausgesetzt.

Literatur:

- GRAMLICH, Günter M.:
Lineare Algebra: Eine Einführung.
5. aktualisierte Aufl. München: Carl Hanser Verlag 2021
- FISCHER, Gerd:
Lernbuch Lineare Algebra und Analytische Geometrie.
4. Aufl. Wiesbaden: Vieweg + Teubner Verlag 2019
- ANTON, Howard:
Elementary Linear Algebra.
John Wiley & Sons Inc 2019

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB379 – Konzepte der funktionalen Programmierung

Verantwortliche:	Torben Tietgen
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB375 – Konzepte der funktionalen Programmierung
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Torben Tietgen

Teilleistung:	TB376 – Übg. Konzepte der funktionalen Programmierung
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	10 Min.
ECTS:	3.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	78 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	93 Stunden
Lehrende:	Malte Heins

Studieninhalte:

- Typen
- Typklassen
- Funktionsdefinitionen
- Rekursive Funktionen
- Funktionen höherer Ordnung
- Korrektheitsargumentationen
- Selbstdefinierte Datenstrukturen und Bäume

Zusätzlich werden in der Übung praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Haskell behandelt, die nicht Bestandteil der Vorlesung sind. Die Bearbeitung der Übungsaufgaben folgt parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen.

Lernergebnisse:

Die Studierenden ...

- kennen die Programmierparadigmen der funktionalen Programmierung und der Algorithmenkonstruktion ohne Programmvariablen, Zuweisungen und Schleifen.
- besitzen das Verständnis und den Umgang mit Rekursion beim Entwickeln von Algorithmen.
- kennen die Vorteile einer streng getypten Programmiersprache bei der Programmentwicklung und beim Erlernen einer neuen Programmiersprache.
- haben das erste Grundverständnis über das Führen von Korrektheitsargumentationen wie Terminierung und Ableitung von Gesetzen und Programmeigenschaften erworben.
- sind geschult bezüglich der Fähigkeiten zur Abstraktion und zur Modellbildung.
- haben das sorgfältige, methodische und systematische Programmieren im Kleinen trainiert.

Voraussetzungen und Empfehlungen:

Grundlegende Programmiererfahrungen in einer beliebigen Programmiersprache sind empfehlenswert, aber nicht vorausgesetzt. Zudem ist die Installation von Haskell auf dem eigenen Rechner empfehlenswert.

Literatur:

- Miran Lipovaca: Learn You a Haskell for Great Good!: A Beginner's Guide, No Starch Press, 2011, ISBN: 978-1593272838
- Graham Hutton: Programming in Haskell, Cambridge University Press, 2016, ISBN: 978-1316626221

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
 - Informatik Bachelor of Science
 - Smart Technology Bachelor of Science
-

◆ MB171 – Seminar Aktuelle technologische Entwicklungen

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB040 – Seminar
Lernform:	Seminar
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	30 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	150 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 165 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.</i>
Lehrende:	Franziska Bönke Gunnar Harms Alexander Fischer Thorsten Giersch Stefan Christoph Weber

Studieninhalte:

Fachvorträge mit anschließender Gruppendiskussion.

Lernergebnisse:

Seminararbeiten dienen insbesondere dem Erlernen von Fertigkeiten zum Erstellen der Bachelor-Thesis. Wesentlich ist die eigenständige Erarbeitung und Darlegung der Inhalte zu einem vorgegebenen Thema unter Einhaltung der Formalia. Die Ausarbeitung soll das Interesse an einer eigenständigen Befassung mit Inhalten aus dem Themengebiet und den Einstieg in die zugehörige wissenschaftliche Fachliteratur und Methodik fördern und anregen. Schließlich ist die obligatorische Präsentation der Ergebnisse ebenfalls Aufgabe innerhalb des Seminars.

Nach erfolgreicher Teilnahme können sie ...

- wissenschaftliche Themen angemessen strukturieren.
- eine eigenständige Zielsetzung erarbeiten und umsetzen.
- Inhalte recherchieren und übersichtlich aufbereiten.
- formale Kriterien sicher beachten und anwenden.
- eine schriftliche Ausarbeitung größeren Umfangs erstellen.
- kontroverse Lehrmeinungen und aktuelle Trends zu einem Thema herausarbeiten.
- ihre Ergebnisse in angemessener Form vortragen und mit den Seminarteilnehmern diskutieren.

Voraussetzungen und Empfehlungen:

Für eine erfolgreiche Teilnahme werden grundlegende wissenschaftliche Arbeitstechniken vorausgesetzt, insbesondere die Fähigkeit zur selbstständigen Literaturrecherche, zur kritischen Lektüre von Fachtexten sowie zur strukturierten Präsentation technologischer Themen.

Literatur:

Recherche nach aufgabenbezogener Literatur, teilweise aufgabenspezifische Vorgabe einzelner Literaturquellen.

Empfehlungen zur Einführung in das wissenschaftliche Arbeiten

- Axel Bänsch, Dorothea Alewell, Wissenschaftliches Arbeiten, 11. Aufl., München [u.a.]: Oldenbourg 2013.
- Werner Heister, Dagmar Weßler-Poßberg, Studieren mit Erfolg: Wissenschaftliches Arbeiten für Wirtschaftswissenschaftler, 2., überarbeitete Auflage, Stuttgart: Schäffer-Poeschel 2011.
- Jens Hiller, Arbeitstechniken und wissenschaftliches Arbeiten, Herne: Kiehl 2017.
- Walter Krämer, Wie schreibe ich eine Seminar- oder Examensarbeit. 3., überarbeitete und aktualisierte Aufl., Frankfurt: Campus 2009.
- Lydia Prexl, Mit digitalen Quellen arbeiten. Richtig zitieren aus Datenbanken, E-Books, YouTube & Co., 2., aktualisierte und erweiterte Aufl., Paderborn: Ferdinand Schöningh (UTB) 2016.
- Manuel René Theisen, Wissenschaftliches Arbeiten: Technik - Methodik Form, 15. Aufl., München: Vahlen 2011.

Studiengänge:

- Smart Technology Bachelor of Science

◆ MB172 – Laborassistentenz

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB206 – Laborassistentenz
Lernform:	Assistenz
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	15 Min.
ECTS:	5,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	35 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	80 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

Im Rahmen der Laborassistentenz werden die Studierenden von den Hochschullehrern mit konkreten (Teil)-Projekten betraut, die sich im Umfeld Organisation und Betreuung der Fachbereichs-Labore bewegen. Diese umfassen nach Maßgabe Wartung und Pflege der vorhandenen Anlagen, Materialbeschaffung und insbesondere auch die Anleitung der Projektgruppen jüngerer Semester im sachgerechten Umgang mit der Laborausstattung. Ebenso in Betracht kommen die Durchführung von Tutorien oder Übungen. Die Laborassistentenz ist selbständig zu bearbeiten und kann die Abstimmung mit anderen Studierenden erfordern.

Lernergebnisse:

Die Studierenden entwickeln unter Anleitung eines Hochschullehrers ...

- die Fähigkeit zur Übernahme von Verantwortung für eine zeitlich begrenzte Aufgabe.
- Kenntnisse über die Organisation, Verwaltung und Betrieb technischer Laboratorien.
- die Fähigkeit zur Weitergabe erlernten Wissens und Fähigkeiten insbesondere in Konzeption, Entwurf, Herstellung und Betrieb smarter Systeme.
- Tiefgehende fachliche Kenntnisse, ausgeprägte Kommunikationsfähigkeit und soziale Kompetenzen.

Voraussetzungen und Empfehlungen:

Für eine erfolgreiche Teilnahme werden solide fachliche Grundkenntnisse im Bereich smarter Systeme sowie die Bereitschaft zur selbstständigen Übernahme von Verantwortung und zur Zusammenarbeit mit anderen Studierenden vorausgesetzt.

Literatur:

aufgabenabhängig

Studiengänge:
• Smart Technology Bachelor of Science

◆ MB257 – Auslandssemester

Verantwortliche:	Herlinde Breckwoldt
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB039 – Auslandssemester
Lernform:	Ausland
Prüfungsform:	Ausland
Prüfungsdauer:	
ECTS:	30,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	25 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	187 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	900 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 1087 Stunden weicht signifikant von erwarteten 30,0 ECTS ab.</i>
Lehrende:	Herlinde Breckwoldt

Studieninhalte:

Für ein freiwilliges Auslandssemester ist der Umfang der zu leistenden ECTS-Punkte (bzw. der gleichwertige Umfang in lokalen Credits) in der jeweiligen Studienordnung vorgegeben. An der ausländischen Hochschule sind fachspezifische Kurse zu belegen, die mit dem in Wedel belegten Studiengang in ergänzendem Zusammenhang stehen. Das Studienprogramm wird vor der Abreise individuell mit dem International Office vereinbart.

Lernergebnisse:

Nach Abschluss des Auslandsemester besitzen die Studierenden ...

- fundierte Sprachkompetenzen in englischer, französischer oder spanischer Sprache.
- erweiterte Kenntnisse über die Kultur des Gastlandes.

Voraussetzungen und Empfehlungen:

Es wird empfohlen, mindestens eine der Sprachen zu beherrschen, die an der ausländischen Hochschule gesprochen wird.

Literatur:

Abhängig von der ausländischen Hochschule

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science

- IT-Management & Consulting Bachelor of Science
 - IT-Management / -Consulting & -Auditing Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB357 – Cross Innovation Class

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch / englisch

Bestandteile:

Teilleistung:	TB351 – Cross Innovation Class
Lernform:	Projektarbeit
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	30 Min.
ECTS:	10,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	45 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	180 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	55 Stunden
Gesamtaufwand:	300 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

Interdisziplinäre Zusammenarbeit von Studierenden unterschiedlicher Fachrichtungen mit Industriepartnern aus verschiedenen Branchen; Leitthema Nachhaltigkeit als übergreifender Rahmen; Phasen des Innovationsprozesses: Kick-off, Strategic Foresight, Feldforschung, Problemdefinition, Ideenfindung, Entwurf und Prototyping; Entwicklung technisch funktionsfähiger Prototypen als Semester-Deliverable; Kommunikation und Koordination an inhaltlichen und organisatorischen Schnittstellen in heterogenen Teams auf basis von offenen Fragestellungen von Industriepartnern.

Lernergebnisse:

Die Studierenden ...

- kennen die wesentlichen Phasen eines nutzerzentrierten Innovationsprozesses (von der Problemdefinition über Ideenfindung bis zum Prototyping) und können diese eigenständig durchlaufen.
- sind in der Lage, aus einer offen gestellten Fragestellung eines Industriepartners eine konkrete, umsetzbare Aufgabenstellung zu entwickeln und diese prototypisch zu realisieren.
- haben praktische Erfahrung in der interdisziplinären Teamarbeit gesammelt und können ihre fachspezifischen Kenntnisse und Fähigkeiten zielgerichtet in eine gemeinsame Gruppenleistung einbringen.
- verstehen die Bedeutung von inhaltlichen und organisatorischen Schnittstellen in branchenübergreifenden Projektkontexten und können diese aktiv gestalten.
- haben ein vertieftes Bewusstsein für Nachhaltigkeitsanforderungen in der Produktentwicklung und können diese als Designkriterien in technische und gestalterische Lösungen integrieren.
- haben Erfahrung im Umgang mit realen Auftraggebern aus der Industrie gesammelt und können Projektergebnisse adressatengerecht präsentieren und kommunizieren.

Voraussetzungen und Empfehlungen:

Für eine erfolgreiche Teilnahme wird die Bereitschaft zur interdisziplinären Zusammenarbeit sowie zur aktiven Mitgestaltung im Rahmen von Gruppenarbeiten vorausgesetzt. Grundlegende fachliche Kenntnisse der eigenen Studienrichtung sind erforderlich, um sinnvoll zur Teamleistung beizutragen. Erfahrungen mit agilen oder designorientierten Arbeitsmethoden sind von Vorteil.

Literatur:

- Brown, T.: *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. HarperBusiness, New York, 2009. ISBN 978-0-06-176608-4.
- Lewrick, M., Link, P., Leifer, L.: *The Design Thinking Playbook*. Wiley, Hoboken, 2018. ISBN 978-1-119-46747-9.
- Osterwalder, A., Pigneur, Y.: *Business Model Generation*. Wiley, Hoboken, 2010. ISBN 978-0-470-87641-1.

Studiengänge:

- Smart Technology Bachelor of Science

◆ MB368 – Venture Lab

Verantwortliche:	Jan-Paul Lüdtkke
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB389 – Venture Lab
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	15 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	45 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	105 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Valena Maiwald

Studieninhalte:

Das Modul gliedert sich in einen wissensvermittelnden und einen anwendungsorientierten Teil.

Ergänzend dazu können die Studierenden erweiterte Angebote in Anspruch nehmen, die fundiertes Wissen für eine angestrebte Gründung vermitteln.

Zu folgenden Themen wird Wissen vermittelt:

- Einführung in Entrepreneurship
- Entstehung innovativer Geschäftsideen
- Zielgruppendefinition und Persona-Methodik
- Value Proposition Design
- Validierung des Problem-Solution-Fits
- Rapid Prototyping
- Business Modell
- Markt und Wettbewerb
- Pitching

Im Rahmen des Moduls werden die Studierenden befähigt, eigenständig und in Begleitung durch Gründungsberatung zu arbeiten und die obengenannten Themen in Teams auf die eigene Gründungsidee zu transferieren und auszuarbeiten. Ergänzt wird dies durch sogenannte Toolboxen, die bei starker Gründungsmotivation besucht werden können. Themen sind dabei zum Beispiel:

- Kalkulation und Preisstrategie
- Startup-Finanzierung und Förderung
- Marketing und Vertrieb
- Rechtsformen und Steuer

Lernergebnisse:

Die Studierenden...

... werden für Entrepreneurship als Karriereweg und Karrierewerkzeug sensibilisiert.

... verstehen den Wert unternehmerischen Denkens in einer Welt, die von hoher Veränderungsgeschwindigkeit, Unsicherheit, Komplexität und Mehrdeutigkeit geprägt ist.

...kennen Werkzeuge zur Identifikation von unternehmerischen Herausforderungen, geschäftlichen Opportunitäten oder bestehenden Problemen bei Zielgruppen.

...können unternehmerische Methoden für die Geschäftsmodellentwicklung identifizierten und anwenden.

...können mit Hilfe von Instrumenten der Lean-Startup-Methode eigene Gründungsvorhaben entsprechend von Zielgruppenbedürfnissen entwickeln, erproben und vor Kunden, Investoren und Partnern gewinnend vorstellen.

... arbeiten interdisziplinär zusammen und erschaffen innovative unternehmerische Vorhaben.

Voraussetzungen und Empfehlungen:

Zusammenhangswissen betriebswirtschaftlicher Grundlagen.

Literatur:

- Freiling, Jörg, Harima, Jan (2019): Entrepreneurship: Gründung und Skalierung von Startups, Gabler Verlag
- Fueglistaller, Urs et al. (2019): Entrepreneurship: Modelle - Umsetzung - Perspektiven Mit Fallbeispielen aus Deutschland, Österreich und der Schweiz, Gabler Verlag
- Hölzle/Tiberius/Surrey (2020): Perspektiven des Entrepreneurships: Unternehmerische Konzepte zwischen Theorie und Praxis
- Kailer/Weiß (2009): Gründungsmanagement kompakt, von der Idee zum Businessplan, Linde Verlag Wien
- Maurya, Ash (2012): Running Lean: Iterate from Plan A to a Plan That Works. Sebastopol/CA 2012
- Osterwalder, Alexander; Pigneur, Yves (2013) Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Hoboken/NJ
- Ries, Eric (2011): The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. New York/NY

Studiengänge:

- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB150 – Bachelor-Thesis

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	BTH – Bachelor-Thesis
Lernform:	Thesis
Prüfungsform:	Abschlussarbeit
Prüfungsdauer:	30 Min.
ECTS:	12,0
Benotung:	Zehntelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	0 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	200 Stunden
Aufwand während Semesterferien:	140 Stunden
Flexibel einteilbarer Aufwand:	20 Stunden
Gesamtaufwand:	360 Stunden
Lehrende:	Sergei Sawitzki

Studieninhalte:

Die Bachelor-Thesis soll im Regelfall in Kooperation mit einem Unternehmen erarbeitet werden. Themen aus den Arbeitsgruppen und Laboren der Hochschule sind ebenfalls möglich. Die Arbeit ist als abschließende, vom Studierenden eigenständig aber hochschul- und unternehmensseitig betreutes Projekt zu verstehen. Im Sinne der Zielsetzung der Bachelor-Ausbildung, der Erlangung des ersten berufsqualifizierenden Abschlusses, ist die Arbeit thematisch an einer Problemstellung eines kooperierenden Unternehmens orientiert oder sie besteht aus einer praxisrelevanten hochschulinternen Aufgabe.

Lernergebnisse:

Die Studierenden ...

- besitzen die Fähigkeit zur Durchführung einer praxisorientierten Arbeit
- können eine Fragestellung selbständig erarbeiten
- können die zu erarbeitende Problematik klar strukturieren
- können die Vorgehensweise und Ergebnisse in einer Ausarbeitung übersichtlich darstellen
- stärken ihre praktischen Fähigkeiten im Projektmanagement-Bereich und zur Selbstorganisation

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und Seminar

Literatur:

themenabhängig

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science

- Computer Games Technology Bachelor of Science
 - Data Science & Artificial Intelligence Bachelor of Science
 - E-Commerce Bachelor of Science
 - Informatik Bachelor of Science
 - IT-Ingenieurwesen Bachelor of Science
 - IT-Management & Consulting Bachelor of Science
 - IT-Management / -Consulting & -Auditing Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB370 – Bachelor-Kolloquium

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB365 – Bachelor-Kolloquium
Lernform:	Kolloquium
Prüfungsform:	Kolloquium
Prüfungsdauer:	30 Min.
ECTS:	3,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	1 Semesterwochenstunden
Termine im Semester:	1 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	10 Stunden
Aufwand während Semesterferien:	10 Stunden
Flexibel einteilbarer Aufwand:	3 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 23 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki

Studieninhalte:

- nach Thema der Bachelor-Arbeit unterschiedlich
- Fachvortrag über das Ergebnis der Bachelor-Arbeit
- Diskussion der Qualität der gewählten Lösung
- Fragen und Diskussion zum Thema der Bachelor-Arbeit und verwandten Gebieten

Lernergebnisse:

Die Studierenden ...

- besitzen die Fähigkeit der konzentrierten Darstellung eines intensiv bearbeiteten Fachthemas.
- verfestigen die Kompetenz, eine fachliche Diskussion über eine Problemlösung und deren Qualität zu führen.
- verfügen über ausgeprägte Kommunikations- und Präsentationsfähigkeiten.

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und Bachelor-Thesis

Literatur:

themenabhängig

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science

- IT-Ingenieurwesen Bachelor of Science
 - IT-Management & Consulting Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB371 – Praktikum

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB366 – Praktikum
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	20 Min.
ECTS:	15,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	0 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	220 Stunden
Aufwand während Semesterferien:	220 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	450 Stunden
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Sammeln von beruflichen Erfahrungen in einem der durch die Prüfungsverfahrensordnung vorgesehenen Tracks:
 - Business-Track, berufliche Tätigkeit in einem etablierten Unternehmen
 - Start-up-Track, Vorbereitung der Gründung eines eigenen Unternehmens
 - Project-Track, Teilnahme an einem größeren Projekt mit wechselnden Projektteams
 - Science-Track, detaillierte und forschungsorientierte Auseinandersetzung mit einem wissenschaftlichen Themenkomplex
- Erstellung eines Praktikumsberichts
- Das berufsbildende Praktikum ist unabhängig vom Track im Umfang von 12 Wochen zu absolvieren

Lernergebnisse:

Die Studierenden

- erweitern ihre sozialen Kompetenzen und ihre Kontakte zu Unternehmen. Beides können sie nach ihrem Studiumsabschluss gewinnbringend für eine Bewerbung oder das Einleben bei ihrem späteren Arbeitgeber bzw. Gründung eines eigenen Unternehmens verwenden
- können Fach- und Methodenkompetenz auf ausgewählte Abläufe und Problemstellungen des betrieblichen Alltags zu übertragen

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und "Soft Skills"

Literatur:

themenabhängig

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science